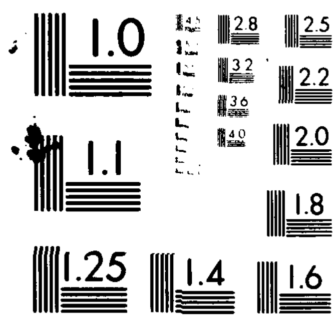
AD-A110 037 LOCKHEED-GEORGIA CO MARIETTA F/G 1/3
NUMERICAL AIRCRAFT DESIGN USING 3-D TRANSONIC ANALYSIS WITH OPT--ETC(U)
AUG 81 P AIDALA F33615-78-C-3014
UNCLASSIFIED LG81ER0107-VOL-3-PT-2 AFWAL-TR-81-3091-VOL-3-PT- NL

END
DATE
FILMED
3 82
DTIC



MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS 1963 A

AD A110037

AFWAL-TR-81-3091
VOLUME III, PART 2

NUMERICAL AIRCRAFT DESIGN USING 3-D TRANSONIC ANALYSIS WITH OPTIMIZATION

VOLUME III
PART 2: USER'S GUIDE TO FIGHTER DESIGN COMPUTER PROGRAM

Lockheed-Georgia Company
86 South Cobb Drive
Marietta, Georgia 30063

Grumman Aerospace Corporation
Bethpage, New York 11714

August 1981

Final Report May 1978 to September 1980

Flight Dynamics Laboratory
Air Force Wright Aeronautical Laboratories
Air Force Systems Command
Wright-Patterson Air Force Base, Ohio 45433

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture use, or sell any patented invention that may in any way be related thereto.

This report has been reviewed by the Office of Public Affairs (ASD/PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

Robert A Large

ROBERT A. LARGE, Capt, USAF
Project Engineer

Lowell C Keel

LOWELL C. KEEL, Maj, USAF
Chief, Aerodynamics & Airframe Branch
Aeromechanics Division

FOR THE COMMANDER

J R Chevalier

JOHN R. CHEVALIER, Colonel, USAF
Chief, Aeromechanics Division

"If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify AFWAL/FIMM, *W-PAFB, OH 45433 to help us maintain a current mailing list".*

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

AIR FORCE/56780/11 December 1981 — 180

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER AFWAL-TR-81-3091 Volume III, Part 2	2. GOVT ACCESSION NO. AD-A110 037
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle) NUMERICAL AIRCRAFT DESIGN USING 3-D TRANSONIC ANALYSIS WITH OPTIMIZATION VOLUME III, PART 2: USER'S GUIDE TO FIGHTER DESIGN COMPUTER PROGRAM	5. TYPE OF REPORT & PERIOD COVERED Final Report May 1978 - September 1980
	6. PERFORMING ORG. REPORT NUMBER LG81ER0107
7. AUTHOR(s) P. Aidala	8. CONTRACT OR GRANT NUMBER(s) F33615-78-C-3014
9. PERFORMING ORGANIZATION NAME AND ADDRESS Grumman Aerospace Corporation Bethpage, New York 11714	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62201F 24041026
11. CONTROLLING OFFICE NAME AND ADDRESS Flight Dynamics Laboratory (AFWAL/FIMM) Air Force Wright Aeronautical Laboratories Wright-Patterson Air Force Base, Ohio 45433	12. REPORT DATE August 1981
	13. NUMBER OF PAGES 75
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report) Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Prepared in cooperation with Lockheed-Georgia Company

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Aircraft	Optimization	Transonic Flow Analysis
Aerodynamics	Wing Design	Numerical Optimization
Canard Configuration	Computational Aerodynamics	
Transonic Flow	Transonic Aircraft Design	

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A User's Guide for the computer code used in the fighter design case study of the Advanced Transonic Technology (ATT) program is presented. The design code includes a 3D transonic wing-body-canard analysis program linked to a numerical optimization routine and a two dimensional strip boundary layer program. The input data required is described in detail and samples of the output are presented.

DD FORM 1473, 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. ABSTRACT (continued)

The purpose of the ATT program was to develop and validate a new transonic wing design procedure using the numerical optimization technique. The new procedure was used to design both a transport and a fighter configuration. Because the missions and design requirements of a fighter and transport are so different, the design procedure was developed along parallel lines. Lockheed-Georgia Co. developed the transport design procedure, and Grumman Aerospace Corp. developed the fighter design procedure.

This document is the second part of a two part volume of detailed User's Guides for the computer programs produced by Lockheed-Georgia Company and Grumman Aerospace Corp. of a new transonic wing design procedure. As in the other volumes of this report, Volume 3 is divided into two parts: Part 1 presents the User's Guide for the transport design programs and Part 2 presents the User's Guide for the fighter design programs.

PREFACE

This document reports results obtained by the Lockheed-Georgia Co. and Grumman Aerospace Corp. under AFFDL Contract # F33615-78-C-3014. The purpose of the contract was to develop and validate a new transonic wing design procedure using the numerical optimization technique. The new procedure was used to design both a transport and a fighter configuration.

Because the missions and design requirements of a fighter and transport are so different, the design procedure was developed along parallel lines. Lockheed-Georgia Co. developed the transport design procedure, and Grumman Aerospace Corp. did the fighter design.

This is Part 2 of a two-part volume: Part 1 details the transport design code Users Guide and Part 2 is the fighter design code Users Guide. There are two other volumes which make up the final report. Volume 1 is an executive summary. Volume 1 is also divided into two parts with Part 1 dealing with the transport design and Part 2 concerned with the fighter design. Volume 2 is a detailed discussion of the technical tasks performed under this contract. Volume 2 is similarly divided into two parts.

Personnel who contributed to this contract effort are: Lockheed-Georgia Company, A. J. Srokowski, M. E. Lores, R. A. Weed and P. R. Smith; Grumman Aerospace Corp., P. Aidala. The authors also wish to acknowledge the assistance given by Capt. R. A. Large who was the AFWAL contract monitor.

CONTENTS

LIST OF ILLUSTRATIONS

LIST OF SYMBOLS

a	Constant coefficients in equation (7).
A, C, D	Constant coefficients in equations (6), (8) and (9).
C_p	Pressure coefficient.
M	Freestream Mach number.
x, X	Streamwise physical coordinate.
y, Y	Spanwise physical coordinate.
z, Z	Vertical physical coordinate.
γ	Specific heat ratio.
ϕ	Perturbation velocity potential.
ξ	Symbolic streamwise computational coordinate.
η	Symbolic spanwise computational coordinate.
ζ	Symbolic vertical computational coordinate.
ξ_∞	Value for ξ in equation (7) that corresponds to upstream and downstream infinity.

<u>SUBSCRIPTS</u>

GLE	Fine grid leading edge.
GTE	Fine grid trailing edge.
x, y, z	Partial derivatives.
n	Normal derivative.

SECTION I

INTRODUCTION

Computational aerodynamic methods for the design of future configurations are becoming more important. New technology concepts and new configurational concepts are being developed to improve transonic performance. Accurate performance predictions are needed to confidently evaluate the benefits of new concepts. In addition, the pursuit of increased performance and reduced development time and costs intensify the use of three-dimensional computational analysis.

The fighter design code developed in this study is a wing-body-canard transonic analysis capability coupled with numerical optimization. The multiple lifting surface transonic analysis capability is the first available for general applications. The wing-body-canard analysis capability was developed from the wing-body code of Boppe [1], while the optimization routines are essentially direct copies of those of Vanderplaats [8,9].

The code has been named PANDORA--Preliminary Automated Numerical Design Of Realistic Aircraft. It was developed on a CDC 7600 computer at NASA Ames Research Center. Computer time for the code development and application was provided through the Applied Computational Aerodynamics (ACA) Branch at Ames. In addition, parts of the work were performed at Ames through the Grumman-Ames Research Associate Program. The code is stored on the Ames central facility 7600 in CDC UPDATE format. Technical details of accessing the code can be provided by the ACA Branch at Ames.

This Users Guide will provide the information needed to use the PANDORA code for configuration analysis or aerodynamic optimization. Volume 2, Part 2 of this final report contains important discussions of the code accuracy and application. Unnecessary details of the code which are the same as described in References [1], [8] or [9] are not repeated here. Those references should be consulted if the particular code details are of interest.

SECTION II

WING-BODY-CANARD TRANSONIC ANALYSIS

The basis computational method employed in the transonic analysis is that of Boppe [1]. An earlier version (1978) of the wing-body analysis code was used as the base on which to develop the wing-body-canard analysis capability. The new developments are described in detail here. The common parts (flow equation, embedded grid interfacing, body modeling, viscous effects) are described briefly. Reference [1] should be consulted if more details are of interest.

1. FLOW EQUATION

The flow equation used in the analysis is an "extended" small-disturbance equation:

$$\begin{aligned}(1-M_\infty^2 - (\gamma+1)M_\infty^2\phi_x - \frac{\gamma+1}{2} M_\infty^2\phi_x^2)\phi_{xx} - 2M_\infty^2\phi_y\phi_{xy} \\ + (1-(\gamma-1)M_\infty^2\phi_x)\phi_{yy} + \phi_{zz} = 0\end{aligned} \tag{1}$$

The additional terms have been added to better capture swept shock waves and more accurately determine the critical velocity. Empirical modifications and similarity variables are not employed.

Pressure coefficients on wing surfaces are computed using the following equation:

$$C_p = -(2\phi_x + (1 - M_\infty^2)\phi_x^2 + \phi_y^2) \qquad (2)$$

To simplify velocity computations on the non-planar body surface, a simplified equation is used:

$$C_p = -(2\phi_x + (1-M_\infty^2)\phi_x^2) \qquad (3)$$

The computational space used in the present method is illustrated in Figure 1. This space is filled with a relatively crude Cartesian mesh. Instead of adopting a far-field solution for the grid outer boundaries, the original x, y, z region is stretched to ξ,η,ζ region in which the boundaries correspond to infinity. The flow field potential is set to zero on all bounding planes except the downstream plane for which the following equation is solved.

$$\phi_{yy} + \phi_{zz} = 0 \qquad (4)$$

The following conditions are enforced at the symmetry plane.

$$\phi_y = 0 \qquad (5a)$$

$$\phi_{xy} = 0 \qquad (5b)$$

The numerical solution of the flow equation uses successive line over-relaxation (SLOR). Vertical line solutions proceed streamwise at a given spanwise plane. For the crude grid solution, the spanwise planes are solved sequentially starting at the most outboard station. For the embedded fine grid solution, the spanwise grid systems are solved sequentially from the wing root to the wing tip and then from the canard root to the canard tip.

2. GRID GENERATION

The key item to enable wing-canard analysis without undue computer storage requirements was the development of suitable grid point distributions. The global crude mesh uses 51, 26 and 31 planes in the streamwise, spanwise and vertical directions, respectively. This results in 41,106 mesh points.

In order to distribute the crude grid streamwise mesh planes as effectively as possible, separate grids are used for wing and wing-canard analysis. Without a canard, the streamwise grid is generated with a tangent function:

$$X = A_1 \mathrm{TAN}((\pi/2)\xi) \qquad (6)$$

This transforms the finite equally-spaced computational domain $-1 \le \xi \le 1$ to the physical region $-\infty \le X \le \infty$. The grid is centered on the wing planform.

When a canard is present, the grid of equation (6) usually results in too few crude grid planes intercepting the canard planform. The streamwise grid transformation used for wing-canard combinations is:

$$X = a_0 + a_1\xi + a_2\xi^2 + a_3\xi^3 + a_4\xi^4 + a_5\xi^5 + a_s\xi/(1-\xi^2) \qquad (7a)$$

$$\xi = \xi/\xi_\infty \qquad (7b)$$

This transforms the finite domain $-1 \leq \xi \leq 1$ to the infinite region $-\infty \leq X \leq \infty$. The constant a_s is empirical, controlling the rate of stretching near infinity. The points at $\xi = \pm 1$ are made the points of maximum density of mesh planes by specifying the second derivative of X to be zero. The value of X and the first derivative of X at $\xi = \pm 1$ are also specified to determine the coefficients a_0 to a_5. The value of ξ_∞ is adjusted iteratively to place approximately fifty percent of the total mesh planes between the most forward and the most aft points on the wing-canard combination.

For an aft swept wing-canard combination, the wing tip and canard tip determine the transformation. The mid-points of the two tip chords are the values of X for $\xi = \pm 1$. The first derivative of X is set to result in a nominal six mesh planes intercepting each of the tip chords. An example of the resulting wing-canard grid transformation and the corresponding physical mesh plane distribution is shown in Figure 3. The mesh generation has been applied to several aft-swept configurations, several forward-swept configurations and to several arbitrary wing-canard parametric variations. Good results were observed in all cases. For forward swept wings, the canard tip and wing tip may be at the same streamwise location. In this case, the two streamwise points used for $\xi = \pm 1$ in the transformation are the wing/canard tip location and near the trailing edge of the centerline of the wing.

The spanwise grid transformation is much simpler:

$$Y = (1/A_2)\mathrm{TANH}^{-1}(\eta) + C_1\eta + C_3\eta^3 \qquad (8)$$

This transforms the domain $0 \leq \zeta \leq 1$ to $0 \leq Y \leq \infty$. The Constant A_2 is determined to place 18 spanwise mesh planes between the centerline and wing tip, with the wing tip falling midway between computational planes. The constants C_1 and C_3 are used to perturb the grid so that the canard tip falls midway between two mesh planes and the wing tip location is not changed. The constants C_1 and C_3 are set to zero when a canard is not present.

The vertical grid transformation is

$$Z = A_3\mathrm{TAN}((\pi/2)\zeta) + D_1\zeta + D_3\zeta^3 \qquad (9)$$

This transforms $-1 \leq \zeta \leq 1$ to $-\infty \leq Z \leq \infty$. The constant A_3 is chosen to provide sufficient mesh stretching for proper supersonic flow development. The constants D_1 and D_3 are set to make the canard plane and a mesh plane coincident, while maintaining a monotonic, positive grid stretching (i.e., $(A_3\pi/2 + D_1) > 0$). When a canard is not present, the constants D_1 and D_3 are set to zero.

3. MULTIPLE WAKE ANALYSIS

With two lifting surfaces and their attendant wakes, the basic algorithm for the SLOR solution was modified. The current capability of the code is an above wing, non-overlapping canard. As sketched in Figure 2, nine distinct flow-field regions for vertical line relaxation occur:

Region 1 - Bounded by top and bottom infinity
Region 2 - Bounded by the canard and top infinity
Region 3 - Bounded by bottom infinity and the canard
Region 4 - Bounded by the canard wake and top infinity
Region 5 - Bounded by bottom infinity and the canard wake
Region 6 - Bounded by the wing and canard wake
Region 7 - Bounded by bottom infinity and the wing
Region 8 - Bounded by the wing wake and the canard wake
Region 9 - Bounded by bottom infinity and the wing wake

A wing alone solution uses only vertical line regions 1 through 5. The potential jump in the wake is constant, equal to that at the trailing edge of the lifting surface. As indicated in Figure 2, the wakes are constrained to remain in the plane of the corresponding lifting surface. This is considered to be consistent with the other small disturbance approximations in the solution, but actually is also done in full potential codes. When the wake of an upstream lifting surface comes arbitrarily close to a second surface, the error of the undeflected wake model may be significant. Unfortunately, detailed pressure data for simple wing-canard geometries does not exist to isolate this effect and evaluate the numerical model. Good force and wing pressure predictions were obtained for the limited analyses down in this study.

4. EMBEDDED FINE GRID SYSTEM

Individual fine grid arrays are constructed for the wing and canard. These secondary mesh systems serve two purposes. First, detailed computations are performed only in a region very close to the surface where gradients are large and details are important. The resulting numerical efficiency permits a very dense computational mesh, a benefit in both the resolution of shock waves and the calculation of configuration forces and moments. Second, the embedded mesh systems are independent and optimized for a particular geometric component. The system is not constrained by a single geometry-fitting transformation. This will facilitate future applications to configurations with multiple components.

Fine grid arrays are set up at each position where a crude spanwise mesh plane cuts the wing and canard surface. With 18 spanwise crude planes between the centerline and wing tip, an isolated wing analysis would have 18 fine wing grids. (A fuselage would reduce the number of fine wing grids by the number of mesh planes within the computational wing root junction.) The number of canard fine grids is proportional to the extent of the canard semispan. If the canard semispan were half that of the wing, then the canard would have approximately half the number of wing fine grids. This would increase the computation time for the fine grid solution by fifty percent. The fine grid arrays are evenly spaced in both the streamwise and vertical directions.

The local section leading edge is placed midway between mesh points and the trailing edge is placed at a mesh point. The vertical mesh spacing is the same for all wing planes. The streamwise mesh spacing is scaled by the local chord. Thus each section has the same number of grid points along the chord. Typically 80 mesh points are placed on the chord, but code dimensions will allow up to 100 points on the chord if greater resolution is desired. The vertical spacing is adjusted to the appropriate value for the canard fine grids. As for the wing, the vertical mesh spacing is the same for all canard planes, and the streamwise spacing is scaled to the local chord. The boundaries of the fine grids are at 20% of the local chord in front of each leading edge and 10% behind each trailing edge. The upper boundary is at 30% of the wing average chord and the lower boundary at 10%. The total number of fine grid points will vary for different canard/wing semispan ratios. The dimensions of the code will allow 108,000 fine grid points. This corresponds to 32 fine grids on the wing and canard, with each having 135 streamwise points and 25 vertical points.

The fine grid system is illustrated in Figure 3. As described above, the fine grids are sheared and tapered to conform to the wing and canard planforms. This represents a transformation function between the fine grid computational domain (ξ,η,ζ) and the physical domain (x,y,z):

$$x = \xi(X_{GTE} - X_{GLE}) + X_{GLE} \tag{10a}$$

$$y = \frac{TANH^{-1}(\eta)}{C_2} \tag{10b}$$

$$z = \zeta \tag{10c}$$

The spanwise (y) transformation is the same as that for the crude grid. The embedded grid approach allows independent transformations for the wing and canard in the streamwise (x) and vertical (z) directions. Thus the sweep and taper of the wing does not effect the canard grid system and the canard sweep and taper does not effect the wing grid system.

The wing fine grid solution need allow for the presence of the canard wake. Figure 4 illustrates the vertical line relaxation regions for the wing fine grid that includes the canard wake. The different regions have their obvious counterpart in the crude grid solution approach (Figure 2). Within an individual fine grid, the vertical line relaxation starts at the upstream boundary and moves to the downstream boundary. The fine grids are solved sequentially, beginning with the wing root. The solution proceeds to the wing tip and then begins the canard solution at the canard root juncture. When the fine grid solution at the canard tip has been updated, the fine grid solution iteration is completed.

As shown in Figure 4, the wing fine grid solution allows only the canard wake - not the canard surface - above the wing surface. The analysis is restricted to non-overlapping, high canard planforms. Strictly, the wing (or canard) surface must not penetrate the canard (or wing) fine grid boundary. A mild violation of this restriction does not disrupt the solution development. A significant violation will lead to rapid divergence of the fine grid solution.

5. FINE/CRUDE GRID INTERFACE

The embedded fine grid interface with the crude grid is accomplished by alternately updating the crude grid and fine grid solutions. Potential values at interior points (i.e., on a lifting surface) of the crude grid are fixed by interpolating the most recent fine grid solution. The potential values at the perimeter of the fine grids are fixed by interpolating the most recent crude grid solution. In order to speed up the overall solution convergence, the fine grid solution is not calculated until the crude grid solution has established the "coarse" characteristics of the flow (typically 75 iterations). Then the crude/fine interaction is begun until both grids are satisfactorily converged (typically an additional 95 cycles). Linear interpolation of the crude grid potential values is used to initialize the fine grid.

One cycle of the crude/fine grid interaction consists of two steps (see Figure 5):

1. The embedded wing grid is swept holding fine grid perimeter points fixed as an outer boundary. Conventional Neumann boundary conditions (ϕ_n) are imposed at fine grid section boundary points forming an inner boundary.

2. The crude grid section boundary points are computed using the potentials at the fine grid section boundary points (linear interpolation). These crude potentials (ϕ) are held fixed for the global crude grid sweep forming an array of Dirichlet inner boundary conditions. Infinity boundary conditions at the limits of the crude computational space form the outer boundary. At the end of the crude grid sweep, crude grid potentials are used to update the fine grid perimeter points.

One solution iteration of only the crude grid takes 1.4 seconds of CDC 7600 CPU time (3.4×10^{-5} CPU seconds per point). One crude/fine cycle takes 4.2 seconds CPU time for 23 wing and canard fine grids. Each fine grid has 120 streamwise points and 25 vertical points. Thus one crude/fine solution cycle solves 110,000 points total (3.8×10^{-5} CPU seconds per point). For different wing-canard combinations, the crude/fine CPU time is essentially proportional to the number of wing and canard fine grid arrays.

The development of the solution is monitored by calculating the maximum update to the flow field potential (DPM in the printout) and its position in the three-dimensional flow field. Typically, for most configurations at flow conditions which are of interest, the value will start near 5×10^{-1} and end near 5×10^{-5} for the final solution. The solution is essentially converged when the circulation (CIR in the printout) shows no significant change in about twenty solution cycles. It is possible that a non-decreasing DPM value is due to "noise" in the crude/fine interface that does not decay. A two or three order of magnitude DPM reduction should be expected. The position of DPM is useful in pinpointing the problem area when convergence is hindered. Problem areas may develop when the flow conditions are extreme or when the geometric representation is in error.

Three other parameters that indicate the solution convergence appear in the code output as DPMAV, RSD and RSDAV. The DPMAV parameter is the average of the absolute value of the flow potential corrections at every point. It is a more reliable indication of the solution convergence than DPM, and is used to satisfy the convergence test input option. The RSD parameter is the largest "error" in the solution. The location of the RSD value is also printed. This "error" is the right hand side of the finite difference equation being solved. It is calculated with both "old" and "new" flow potential values. The RSDAV parameter is the average of the absolute value of the error at every point.

The indices for the location of DPM and RSD are shifted when the point is located at the lower side of the wing or canard plane. The jump in potential at these locations is handled by an extra array of potential values that have their own index system. In the crude grid, 500 is added to the streamwise index for the canard plane lower side location and 500 + IMAX is added to the streamwise index for the wing plane lower side location (the default for IMAX is 51). In the fine grid system, 500 is added to the streamwise index for the plane of the lifting surface (wing in a wing fine grid, canard in a canard fine grid) and 500 + IMAXW is added to the streamwise index for a canard wake above the wing surface (the default for IMAXW is 120).

No precise non-dimensional parameter is available to compare the solution convergence for different grids. The DPM and DPMAV values are scaled by the wing average cord. This results in the same values for DPM and DPMAV if the scale of the input geometry is changed (e.g., full scale or model scale). Changes to the planform (e.g., aspect ratio or taper) would result in DPM and DPMAV values that are not directly comparable. As mentioned above, the reduction in DPM and DPMAV relative to the starting value should be used to evaluate the solution convergence.

6. BOUNDARY LAYER CALCULATION

Viscous effects are computed in the analysis code by coupling a modified Bradshaw boundary layer computation with the inviscid potential flow solution. The boundary layer calculation is virtually identical to the method developed by Mason [2]. The method employs the modified chord technique of Nash [3], which represents an infinite sheared wing boundary layer calculation. The wing sweep angle is that of the local mid-chord span line, such that it may vary across the span. The 2-D Bradshaw turbulent boundary layer analysis [4] provides the foundation for the method. The use of a modified 2-D boundary layer analysis greatly reduces the necessary computer time and has demonstrated good results for several different codes [1,2,5].

The boundary layer calculation provides a displacement thickness and skin friction calculation at each analysis station of the wing and canard. The slope of the displacement thickness is used to modify the surface boundary conditions in the inviscid solution. The local skin friction calculation is used to provide a viscous drag estimate for the configuration at the end of the analysis run.

Surface pressures are calculated from the fine grid potential solution every 20 cycles. The boundary layer calculation for twenty three wing-canard stations (an upper and lower surface calculation at each station) takes approximately 22 seconds of CDC 7600 CPU time. Thus four boundary layer calculations during the inviscid solution with a boundary layer calculation at the end will add 110 seconds of CPU time to that for the inviscid solution alone. As mentioned for the inviscid solution, the viscous/inviscid interaction is essentially converged when the change in lift (circulation) is negligible.

7. FUSELAGE MODELING

The body is modeled in the solution by a constant cross-section computational surface in the Cartesian crude grid. The input data allow for simple axisymmetric body definition or detailed "Quick Geometry" [6,7] body definition. Body boundary conditions are imposed by fixing the velocity potential values on the body computational surface. The procedure follows that of Reference [2] and is described in detail in Reference [1]. Body pressures from equation (3) are used to produce a calculation of the body force and moment contribution.

SECTION III

COPES/CONMIN OPTIMIZATION

The wing-body-canard analysis code was coupled with the COPES and CONMIN routines of Vanderplaats [8,9]. The COPES code is a control program that connects the numerical optimization code CONMIN with the aerodynamic analysis code. The structure of the PANDORA code allows the numerical optimization to be coupled to any analysis code. Several "analysis" codes could be coupled together to provide information for the optimization. This information might be supersonic performance, TOGW changes or mechanical system sizing estimates. The available computer resources represent the only limit of complexity.

The COPES and CONMIN routines were slightly modified for inclusion in the PANDORA code. Changes were made in the main COPES routine (Program COPES in Overlay(0,0)) and subroutine CNMN06 (in Overlay(5,0)). The change to CNMN06 is important. It provides subroutine ANALIZ with the index of the best result during an optimization search. In this way, the next search solution can be restarted from the best previous result. This more closely models the search strategy in CONMIN, resulting in more accurate information from the aerodynamic analysis. The changes in the COPES and CONMIN routines are identified in the code with comment cards.

The computer code is written in FORTRAN, employing the CDC overlay structure. Storage requirements on a CDC 7600 are $140K_8$ small core memory and $770K_8$ large core memory. Typical CPU times on the 7600 for a wing-body-canard analysis with viscous effects is eleven minutes. The CPU time for an optimization run will vary according to the complexity of the optimization problem, as discussed below.

The optimization algorithm within CONMIN is a modified Method of Feasible Directions [9]. The gradient information for the algorithm is calculated by sequentially perturbing each design variable. Each design variable perturbation requires analysis by the flow solution routines. Thus, the computer time is

proportional to the number of design variables. The gradient information establishes a search direction that should improve the design (decrease the objective function) while satisfying any constraints. The search direction is "explored" until a relative optimum is found or any constraints become violated. If the starting conditions violated any constraints, the search direction will be that which satisfies the constraints with the least objective function increase. One to four flow solutions are required during the search. Completion of the search constitutes one optimization iteration. The following discussion of COPES usage is excerpted from Reference [8]. A more detailed description will be found in the reference. Reference [9] should be consulted if more details of the numerical optimization algorithm is desired.

If it is desired to run only a simple analysis using COPES, only three data cards are required for the COPES program: a TITLE card, a control parameter, (NCALC = 1), and an END card. If the optimization or parametric analysis (sensitivity) capabilities of COPES are to be used, additional data must be read. This data will identify which parameters in the global common block, GLOBCM, are used. To set up the COPES data, the user must have a basic understanding of how the data in the global common block is accessed by COPES. This is outlined in the following section.

1. OPTIMIZATION DATA MANAGEMENT

In order to perform design operations, the COPES program must access the data in common block GLOBCM. This done by defining the location in GLOBCM where a specified parameter resides. For example, consider the common block for a cantilevered beam design problem:

```
COMMON/GLOBCM/B,H,VOL,BSTRES,SHRSTR,DELTA,HB,E,AL
```

The volume of material, VOL, is the third parameter in the common block; that is, it resides in location 3, referred to as the global location number. Similarly the bending stress, BSTRES, is in global location 4 and the beam width is in global location 1. Thus, the parameters are referred to by their respective location numbers in global common.

For convenience in preparing data for the COPES program, a simple "CATALOG" of parameters may be defined. For the cantilevered beam, this catalog would be:

GLOBAL LOCATION	FORTRAN NAME	DEFINITION
1	B	Beam width
2	H	Beam height
3	VOL	Volume of material
4	BSTRES	Maximum bending stress
5	SHRSTR	Maximum shear stress
6	DELTA	Deflection under the load
7	HB	Ratio, H/B
8	E	Young's modulus
9	AL	Length of beam

As another example, consider a global common block containing arrays:

GLOBAL LOCATION	FORTRAN NAME	DEFINITION
1	A	Area
2	Y(10)	Vector_
12	Q	.
13	C(2,2)	.
17	H	etc.

The dimensions are given with the FORTRAN name as a reminder that the parameter is an array. In this case, the third parameter in the Y array is in global location 4. Remembering that arrays are stored column by column the C(1,2) array location is in global location 15.

It will be seen that identifying parameters according to their location in GLOBCM provides a great deal of flexibility in using the COPES program for design.

In the following section, definitions of terms commonly used in automated design are given for easy reference.

2. OPTIMIZATION TERMINOLOGY

The COPES program currently provides six specific capabilities:

1. Simple analysis, just as if COPES was not used.

2. Optimization - Minimization or maximization of one calculated function with limits imposed on other functions.

3. Sensitivity analysis - the effect of changing one or more design variables on one or more calculated functions.

4. Two-variable functions space - analysis for all specified combinations of two design variables.

5. Optimum sensitivity - same as sensitivity analysis except, at each step, the design is optimized with respect to the remaining independent design variables.

6. Approximate optimization - optimization using approximation techniques. Usually more efficient than standard optimization for up to 10 design variables of if multiple optimizations are to be performed.

In defining the data required to execute the COPES program, the following definitions are useful.

Design Variables - Those parameters which the optimization program is allowed to change in order to improve the design. Design variables appear only on the right hand side of equations in the analysis program. COPES considers two types of design variables, independent and dependent. If two or more variables are always required to have the same value or be in a constant ratio, one is the independent variable while the remaining are dependent variables. For example, if the height is required to be 10 times the width of the cantilevered beam, B would be the independent variable while H would be the dependent variable.

Objective Function - The parameter which is to be minimized or maximized during optimization. Also the parameters calculated as functions of specified design variables during a sensitivity or two-variable function space study. Objective functions always occur on the left side of equations, unless the objective function is also a design variable (the beam height may be minimized as an objective function if it is also a design variable. In this way, the minimum height is found for which no constraints are violated). An objective function may be linear or non-linear, implicit or explicit, but must be a function of the design variables to be meaningful.

Constraint - Any parameter which must not exceed specified bounds for the design to be acceptable. Constraint functions always appear on the left side of equations. Just as for objective functions, constraints may be linear or non-linear, implicit or explicit, but must be functions of the design variables.

Constraint Set - A group of constraints which appear consecutively in the global common block and which all have the same limits imposed. This is a convenience which allows several constraints to be identified with a minimum of data.

Global Common - Common block GLOBCM containing design information.

Global Location - Location of a particular parameter in GLOBCM.

3. OPTIMIZATION INPUT DATA FORMAT

In order to execute the COPES program it is necessary to provide formatted data for COPES, followed by data for the ANALIZ program which is coupled to COPES. Section 4.1 defines the data which is required by COPES. The data is segmented into "BLOCKS" for convenience. All formats are alphanumeric for TITLE, and END cards, F10 for real data and I10 for integer data.

The COPES data begins with a TITLE card and ends with an END card. This is followed by data to be read by the user supplied subroutine ANALIZ.

Comment cards may be inserted anywhere in the COPES data stack prior to the END card, and are identified by a dollar sign ($) in column 1.

While the input description defines COPES data in formatted fields of ten, the data may actually be read in more conveniently by separating data by commas or one or more blanks. If more than one number is contained on an unformatted data card, a comma must appear somewhere on the card. If exponential numbers such as 2.5+10 are read on an unformatted card, there must be no embedded blanks. Unformatted cards may be intermingled with formatted cards. Real numbers on an unformatted card must have a decimal point.

Examples

Unformatted data:

```
5,7,1.3,1.0+20,0,-5.1
5,7,1.3,1.0+20,,-5.1
5 7 1.3 1.0+20,, - 5.1
5    7  1.3, 1.0+20 0 -5.1
```

Equivalent formatted data:

col	10	20	30	40	50	60	70	80
	5	7	1.3	1.0+20	0	-5.1		

Unformatted data

```
2
2,3
2 3
```

Equivalent formatted data:

col	10	20	30
	2		
	2	3	

2 3

Note: These data contain no commas, so it is assumed to be formatted already.

Unformatted data:

```
1,2,3,4,5,6,7,8,9,10,11
```

col	10	20	30	40	50	60	70	80
	1	2	3	4	5	6	7	8
	9	10	11					

Note that two formatted data cards are created here. Unformatted data:

```
1,2,3,4,5,6
7,8,9,10,11
```

Equivalent formatted data:

col	10	20	30	40	50	60	70	80
	1	2	3	4	5	6		
	7	8	9	10	11			

Note that the above two examples do not produce the same formatted data cards.

SECTION IV

INPUT DATA DESCRIPTION

This section describes the necessary solution/optimization parameters and geometry input needed to operate the PANDORA code. The information of Section III is important for using the optimization. Only the one-cycle analysis (NCALC = 1 in DATA BLOCK B) and optimization (NCALC = 2) COPES options have been used in this effort. The transonic analysis input is an extension of that from the base code [1]. A set of Usage Notes appears at the end of the analysis input description to elaborate on the less familiar input parameters. The reader is reminded that Volume 2, Part 2 of this final report contains important discussions of the code application and accuracy.

1. OPTIMIZING INPUT DATA

DATA BLOCK A

DESCRIPTION: Title card.

FORMAT AND EXAMPLE

	1	2	3	4	5	6	7	8	FORMAT
TITLE									20A4

CANTILEVERED BEAM DESIGN

FIELD	CONTENTS
1-8	Any 80 character title may be given on this card.

DATA BLOCK B

DESCRIPTION: Program Control Parameters.

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	FORMAT
NCALC	NDV	NSV	N2VAR	NXAPRX	IPNPUT	IPDBG	7I10
2	2	3	5	2	0	0	

FIELD			CONTENTS
1	NCALC:		Calculation Control
	0	-	Read input and stop. Data of blocks A, B and V is required. Remaining data is optional.
	1	-	One cycle through program. The same as executing ANALIZ stand-along (i.e., no optimization). Data of blocks A, B and V is required. Remaining data is optional.
	2	-	Optimization. Data of blocks A-I and V is required. Remaining data is optional.
	3	-	Sensitivity analysis. Data of blocks A, B, P, Q and V is required. Remaining data is optional.
	4	-	Two variable function space. Data of blocks A, B, and R-V is required. Remaining data is optional.
	5	-	Optimum Sensitivity. Data of blocks A-I, P, Q, and V is required. Remaining data is optional.
	6	-	Optimization using approximation techniques. Data of blocks A-O and V is required. Remaining data is optional
2	NDV:		Number of design variables on which sensitivity analysis or optimization will be performed.
4	N2VAR:		Number of objective functions in a two variable function space study.
5	NXAPRX:		Number of X-variables for approximate analysis/optimization. $\geq$ NDV
6	IPNPUT:		Input print control
	0	-	Print card images of data plus formatted print of input data
	1	-	Formatted print only of input data
	2	-	No print of input data.
7	IPDBG:		Debug print control
	0	-	Off
	1	-	On, ANALIZ called for output after each analysis.

DATA BLOCK C OMIT IF NDV = 0 IN BLOCK B

DESCRIPTION: Integer optimization control parameter.

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	8	FORMAT
IPRINT	ITMAX	ICNDIR	NSCAL	ITRM	LINOBJ	NACMX1	NFDG	7I10
5	0	0	0	0	0	0	0	

FIELD		CONTENTS
1	IPRINT:	Print control used in the optimization program CONMIN.
	0 -	No print during optimization.
	1 -	Print initial and final optimization information.
	2 -	Print above plus objective function value and design variable values at each iteration.
	3 -	Print above plus constraint values, direction vector and move parameter at each iteration.
	4 -	Print above plus gradient information.
	5 -	Print above plus each proposed design vector, objective function and constraint values during the one-dimensional search.
2	ITMAX:	Maximum number of optimization iterations allowed. DEFAULT = 20.
3	ICNDIR:	Conjugate direction restart parameter. DEFAULT = NDV + 1.
4	NSCAL:	Scaling parameter. .GT.0 - Scale design variables to order of magnitudes one every NSCAL iterations. .LT.0 - Scale design variables according to user-input scaling values. Good value is ICNDIR, 1 is <u>not</u>.
5	ITRM:	Number of consecutive iterations which must satisfy relative or absolute convergence criterion before optimization process is terminated. DEFAULT = 3.
6	LINOBJ:	Linear objective function identifier. If the optimization objective is known to be a linear function of the design variables, set LINOBJ = 1. DEFAULT = Non-linear.
7	NACMX1:	One plus the maximum number of active constraints anticipated. DEFAULT = NDV + 2.

FIELD (cont'd)			CONTENTS (cont'd)
8	NFDG:		Finite difference gradient identifier.
	0	-	All gradient information is computer by finite difference with CONMIN.
	1	-	All gradient information is computed analytically by the user-supplier code.
	2	-	Gradient of objective is computed analytically. Gradients of constraints are computed by finite difference within CONMIN.

REMARKS

1) Currently NFDG must be zero in COPES.
2) IPRINT = 5 is recommended.

DATA BLOCK D OMIT IF NDV = 0 IN BLOCK B

DESCRIPTION: Floating point optimization program parameters.

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	FORMAT
FDCH	FDCHM	CT	CTMIN	CTL	CTLMIN	THETA	7F10
0.0	0.0	0.0	0.0	0.0	0.0	0.0	
DELFUN	DABFUN	ALPHAX	ABOBJ1				
0.0	0.0	0.0	0.0				

NOTE: TWO CARDS ARE READ HERE.

FIELD		CONTENTS
1	FDCH:	Relative change in design variables in calculating finite difference gradients. DEFAULT = 0.01
2	FDCHM:	Minimum absolute step in finite difference gradient calculations. DEFAULT = 0.001.
3	CT:	Constraint thick parameter. DEFAULT = 0.05.
4	CTMIN:	Minimum absolute value of CT considered in the optimization process. DEFAULT = 0.004.
5	CTL:	Constraint thickness parameter for linear constraints. DEFAULT = -0.01.
6	CTLMIN:	Minimum absolute value of CTL considered in the optimization process. DEFAULT = 0.001.
7	THETA:	Mean value of push-off factor in the method of feasible directions. DEFAULT = 1.0.
1	DELFUN:	Minimum relative change in objective function to indicate convergence of the optimization process. DEFAULT = 0.001.
2	DABFUN:	Minimum absolute change in objective function to indicate convergence of the optimization process. DEFAULT = 0.001 times the initial objective value.
3	ALPHAX:	Maximum fractional change in any design variable for first estimate of the step in the one-dimensonal search. DEFAULT = 0.1.

FIELD (cont'd) CONTENTS (cont'd)

4	ABOBJ1:	Expected fractional change in the objective function for first estimate of the step in the one-dimensional search. DEFAULT = 0.1

REMARKS

1) The DEFAULT values for these parameters usually work well.

DATA BLOCK E OMIT IF NDV = 0 IN BLOCK B

DESCRIPTION: Total number of design variables, design objective identification and sign.

FORMAT AND EXAMPLE

1	2	3	FORMAT
NDVTOT	IOBJ	SGNOPT	2I10,F10
0	3	-1.0	

FIELD		CONTENTS
1	NDVTOT:	Total number of variables linked to the design variables. This option allows two or more parameters to be assigned to a single design variable. The value of each parameter is the value of the design variable times a multiplier, which may be different for each parameter. DEFAULT = NDV.
2	IOBJ:	Global variable location associated with the objective function in optimization.
3	SGNOPT:	Sign used to identify whether function is to be maximized or minimized. +1.0 indicates maximization. -1.0 indicates minimization. If SGNOPT is not unity in magnitude it scales the magnitude of the objective.

DATA BLOCK F OMIT IF NDV = 0 IN BLOCK B

DESCRIPTION: Design variable bounds, initial values and scaling factors.

FORMAT AND EXAMPLE

1	2	3	4	FORMAT
VLB	VUB	X	SCAL	4F10
.5	5.	0.0	0.0	

NOTE: READ ONE CARD FOR EACH OF THE NDV INDEPENDENT DESIGN VARIABLES. VALUES ARE IN ORDER FOR DESIGN VARIABLES NDSGN SEQUENCE (I.E., INPUT SEQUENCE)

FIELD		CONTENTS
1	VLB:	Lower bound on the design variable. If VLB.LT.-1.0E+15, no lower bound.
2	VUB:	Upper bound on the design variable. If VUB.GT.10.E+15, no upper bound.
3	X:	Initial value of the design variable. If X is non-aero, this will supercede the value initialized by the user-supplied subroutine ANALIZ.
4	SCAL:	Design variable scale factor. Not used if NSCAL.GE.0 in BLOCK C.

DATA BLOCK G OMIT IF NDV = 0 IN BLOCK B

DESCRIPTION: Design variable identification.

FORMAT AND EXAMPLE

1	2	3	FORMAT
NDSGN	IDSGN	AMULT	2I10,F10
1	1	1.0	

NOTE: READ ONE CARD FOR EACH OF THE NDVTOT DESIGN VARIABLES.

FIELD		CONTENTS
1	NDSGN:	Design variable number associated with this variable.
2	IDSGN:	Global variable number associated with this variable.
3	AMULT:	Constant multiplier on this variable. The value of the variable will be the value of the design variable, NDSGN, times AMULT. DEFAULT = 1.0

DATA BLOCK H OMIT IF NDV = 0 IN BLOCK B

DESCRIPTION: Number of constrained parameters.

FORMAT AND EXAMPLE

1	FORMAT
NCONS	I10
4	

FIELD		CONTENTS
1	NCONS:	Number of constraint sets in the optimization problem.

REMARKS

1) If two or more adjacent parameters in the global common block have the same limits imposed, these are part of the same constraint set.

DATA BLOCK I OMIT IF NDV = 0 IN BLOCK B, OR NCONS = 0 IN BLOCK H

DESCRIPTION: Constraint identification and constraint bounds.

FORMAT AND EXAMPLE

1	2	3	4	FORMAT
ICON	JCON	LCON		3I10
4	0	0		
BL	SCAL1	BU	SCAL2	
-1.0+20	0.0	20000.	0.0	

NOTE: READ TWO CARDS FOR EACH OF THE NCONS CONSTRAINT SETS.

FIELD		CONTENTS
1	ICON:	First global number corresponding to the constraint set.
2	JCON:	Last global number corresponding to the constraint set. DEFAULT = ICON.
3	LCON:	Linear constraint identifier for this constraint set. LCON = 1 indicates linear constraints. If in doubt, use non-linear.
1	BL:	Lower bound on the constrained variables. If BL.LT. -1.0E+15, no lower bound.
2	SCAL1:	Normalization factor on lower bound. DEFAULT = MAX of (ABS(BL), 0.1).
3	BU:	Upper bound on the constrained variables. If BU.GT.1.0E+15, no upper bound.
4	SCAL2:	Normalization factor on upper bound. DEFAULT = MAX of (ABS(BU), 0.1).

REMARKS

1) The normalization factor should usually be defaulted.

2) The constraint functions sent to CONMIN are of the form: (BL - VALUE)/ SCAL1 .LE. 0.0 and (VALUE - BU)/SCAL2 .LE. 0.0.

3) Each constrained parameter is converted to two constraints in CONMIN unless ABS(BL) or ABS(BU) exceeds 1.0E+15, in which case no constraint is created for that bound.

DATA BLOCK J OMIT IF NXAPRX = 0 IN BLOCK B

DESCRIPTION: Approximate analysis/optimization control parameters.

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	8	FORMAT
NF	NXS	NXFS	NXA	INOM	ISCRX	IXCRXF	IPAPRX	8I10
5	1	1	1	0	0	0	1	
KMIN	KMAX	NPMAX	JNOM	INXLOC	INFLOC			6I10
0	0	0	0	0	0			

FIELD		CONTENTS
1	NF:	Number of functions to be approximated. DEFAULT = number of optimization objective and constraint functions.
2	NXS:	Number of X-vectors read as data.
3	NXFS:	Number of X-F pairs read as data.
4	NXA:	If non-zero, the design variables read by SUBROUTINE ANALIZ form an X-vector.
5	INOM:	Nominal X-vector about which to do Taylor expansion. DEFAULT = best available.
6	ISCRX:	File from which NXS X-vectors are read. DEFAULT = 5.
7	IXCRXF:	File from which NXFS X-F pairs of data are read. DEFAULT = 5.
8	IPAPRX:	Print control, 1 to 4. 4 is most.
1	KMIN:	Minimum number of approximation iterations.
2	KMAX:	Maximum number of approximation iterations.
3	NPMAX:	Maximum number of designs retained for Taylor series expansion.
4	JNOM:	Number of iterations after which the best design is picked as nominal.

FIELD (cont'd)		CONTENTS (cont'd)
5	INXLOC:	X-variable global location identifier. If INXLOC = 0, the Taylor series expansion is on the design variables listed in BLOCK G.
6	INFLOC:	Function global location identifier. If INFLOC = 0, the Objective and constraint functions identified in BLOCKS E and I are the functions on which the Taylor series expansion is performed.

REMARKS

1) If ISCRX and/or ISCRXF file number is other than 5, the data read from that file is assumed to be binary data.

2) If NXS = NSFS = 0, NXA is defaulted to NXA = 1, even if it is read as zero. Also, a second vector of design variables is automatically defined by COPES to yield two independent designs to start the optimization.

DATA BLOCK K OMIT IF NDV = 0 IN BLOCK B, OR NXAPRX = 0 IN BLOCK B

DESCRIPTION: Bounds and multipliers for approximate optimization.

FORMAT AND EXAMPLE

```
       1        2    3    4    5    6    7    8     FORMAT
      DX1      DX2  DX3  DX4  DX5  ...  ...  ...       8F10
    .5       2.
     XFACT1  XFACT2                                  2F10
    0.       0.
```

NOTE: TWO OR MORE CARDS ARE READ HERE

FIELD		CONTENTS
1-8	DXI:	Allowable change (in magnitude) of the Ith design variable during each approximate optimization.
1	XFACT1:	Multiplier on DXI when the diagonal elements of the H matrix are available. DEFAULT = 1.5.
2	XFACT2:	Multiplier on DFXI when all elements of the H matrix are available. DEFAULT = 2.0.

DATA BLOCK L OMIT IF NXAPRX = 0 IN BLOCK B OR INXLOC = 0 IN BLOCK J

DESCRIPTION: GLOBAL LOCATIONS OF APPROXIMATING VARIABLES.

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	8	FORMAT
LOCX1	LOCX2	LOCX3	LOCX4	.....	.....	.....	.....	8I10
1	2							

NOTE: MORE THAN ONE CARD MAY BE READ HERE.

FIELD		CONTENTS
1-8	LOCI:	Global location of Ith approximating variable.

REMARKS

1) If INXLOC = 0 in BLOCK J, this data is not read. In this case, the data is defaulted to be the global locations of the design variables (IDSGN values in BLOCK G).

DATA BLOCK M OMIT IF NXAPRX = 0 IN BLOCK B OR INFLOC = 0 IN BLOCK J

DESCRIPTION: Global locations of functions to be approximated.

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	8	FORMAT
LOCF1	LOCF2	LOCF3	LOCF4	.....	.....	.....	.....	8I10
3	5	6	4					

NOTE: MORE THAN ONE CARD MAY BE READ HERE.

FIELD		CONTENTS
1-8	LOCI:	Global location of Ith function to be approximated.

REMARKS

1) If INFLOX = 0 in BLOCK J, this data is not read. In this case, the data is defaulted to be the global locations of the objective function (IOBJ in BLOCK E) followed by the global locations of the constrained parameters (ICON, JCON in BLOCK I).

DATA BLOCK N OMIT IF NXS = 0 IN BLOCK J

DESCRIPTION: X-vectors for approximate optimization.

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	8	FORMAT
XI1	XI2	XI3	XI4	...	...	...	...	8F10
4.	15.							

NOTE: NXS SETS OF DATA ARE READ HERE.

NOTE: MORE THAN ONE CARD MAY BE READ FOR EACH SET OF DATA.

FIELD		CONTENTS
1-8	XIJ:	Jth value of Ith X-vector, J = 1,NXAPRX.

DATA BLOCK O OMIT OF NXFS = 0 IN BLOCKS J

DESCRIPTION: X-F pairs of information for approximate optimization.

FORMAT AND EXAMPLE

```
     1        2         3         4        5      6      7      8       FORMAT
    X1       X2        X3        X4       ...    ...    ...    ...        8F10
2.       18.
    Y1       Y2        Y3        Y4       Y5     ...    ...    ...
7200.    416.667   .914495   18418.419
```

NOTE: NSFS SETS OF DATA ARE READ HERE.

NOTE: MORE THAN ONE CARD MAY BE REQUIRED FOR XI OR YI.

NOTE: NXAPRX VALUES OF X AND NF VALUES OF Y ARE READ FOR EACH SET OF DATA.

NOTE: MANY SIGNIFICANT DIGITS ARE DESIRABLE.

FIELD		CONTENTS
1-8	XI:	Ith value of X, I = 1,NXAPRX.
1-8	YI:	Ith value of Y, I = 1,NF.

DATA BLOCK P OMIT IF NSV = 0 IN BLOCK B

DESCRIPTION: Sensitivity objectives (function values).

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	8	FORMAT
NSOBJ	IPSENS							2I10
5	0							
NSN1	NSN2	NSN3	NSN4	NSN5	...	...	...	8I10
3	4	5	6	7				

NOTE: TWO OR MORE CARDS ARE READ HERE.

FIELD		CONTENTS
1	NSOBJ:	Number of separate objective functions to be calculated as functions of the sensitivity variables.
2	IPSENS:	Print control. If IPSENS.GT.0, detailed print will be called at each step in the sensitivity analysis. DEFAULT = No print.
1-8	NSNI:	Global variable number associated with the sensitivity objective functions.

REMARKS

1) More than eight sensitivity objectives are allowed. Add data cards as required to contain data.

DATA BLOCK Q OMIT IF NSV = 0 IN BLOCK B

DESCRIPTION: Sensitivity variables.

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	8	FORMAT
ISENS	NSENS							2I10
9	4							
SNS1	SNS2	SNS3	SNS4	....	....	....	....	8F10
200.	100.	150.	250.					

NOTE: READ ONE SET OF DATA FOR EACH OF THE NSV SENSITIVITY VARIABLES.

NOTE: TWO OR MORE CARDS ARE READ FOR EACH SET OF DATA.

FIELD		CONTENTS
1	ISENS:	Global variable number associated with the sensitivity variable.
2	NSENS:	Number of values of this sensitivity variable to be read on the next card.
1-8	SENSI:	Values of the sensitivity variable. I = 1,NSENS. I = 1 correspond to the nominal value.

REMARKS

1) More than eight values of the sensitivity variable are allowed. Add data cards as required to contain the data.

DATA BLOCK R OMIT IT N2VAR = 0 IN BLOCK B

DESCRIPTION: Two variable function space control parameters.

FORMAT AND EXAMPLE

1	2	3	4	5	FORMAT
N2VX	M2VX	N2VY	M2VY	IP2VAR	5I10
1	4	2	5	0	

FIELD		CONTENTS
1	N2VX:	Global location of the X-variable in the two variable function space.
2	M2VX:	Number of values of X to be considered.
3	N2VY:	Global location of the Y-variable in the two variable function space.
4	M2VY:	Number of values of Y to be considered.
5	IP2VAR:	Print control. If IP2VAR.GT.0, detailed print will be called at each step (each X-Y combination). DEFAULT = No print.

DATA BLOCK S OMIT IF N2VAR = 0 IN BLOCK B

DESCRIPTION: Objective functions of the two variable function space study.

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	8	FORMAT
NZ1	NZ2	NZ3	NZ4	NZ5	...	...	...	8I10
3	4	5	6	7				

FIELD		CONTENTS
1-8	NZI:	Global location corresponding to the Ith function of X and Y to be calculated. N2VAR values are read here.

REMARKS

1) More than eight objective functions are allowed. Add data cards as required to contain the data.

DATA BLOCK T OMIT IF N2VAR = 0 IN BLOCK B

DESCRIPTION: Values of the X-variable in a two variable function space study.

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	8	FORMAT
X1	X2	X3	X4	...	...	...	...	8F10
0.5	1.0	1.5	2.0					

FIELD		CONTENTS
1-8	XI:	Values of the X-variable in the two variable function space. M2VX values are read here.

REMARKS

1) More than eight values are allowed. Add data cards as required to contain the data.

DATA BLOCK U OMIT IF N2VAR = 0 IN BLOCK B

DESCRIPTION: Values of the Y-variable in a two variable function space study.

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	8	FORMAT
Y1	Y2	Y3	Y4	Y5	...	...	...	8F10
4.0	8.0	12.0	16.0	20.0				

FIELD		CONTENTS
1-8	YI:	Values of the Y-variable in the two variable function space. N2VY values are read here.

REMARKS

1) More than eight values are allowed. Add data cards as required to contain the data.

DATA BLOCK V

DESCRIPTION: COPES data 'END' card.

FORMAT AND EXAMPLE

```
            1              FORMAT
           END               3A1
        END
```

FIELD	CONTENTS
1	The word 'END' in colums 1-3

REMARKS

1) This card MUST appear at the end of the COPES data.

2) This ends the COPES input data.

3) Data for the user-supplied routine, ANALIZ, follows this.

2. TRANSONIC ANALYSIS INPUT DATA

NOTE: Excluding literal cards, all input data cards are 7F10. format.

CARD NUMBER	CARD COLUMN	VARIABLE NAME	DESCRIPTION
Card 1-A	1-80	TTLE	Configuration or run title to identify graphic and printed output.
Card 2-A	1-10	CASE	= 1. Isolated Body, used for input check of complex body definition. No flow solution. (Omit cards 3-A, 4-A and all cards -C, -W).
			= 2. Isolated Wing (omit all cards -C, -B).
			= 3. Wing-Body (omit cards -C).
			= 4. Isolated wing-canard (omit all cards -B).
			= 5. Wing-Body-Canard.
	11-20	AMACH	Mach Number (AMACH < 1.0).
	21-30	AOA	Angle-of-Attack (degrees).
	31-40	WPO	≤-3. Same as WPO = -2. plus omit grid and body information output.
			≤-2. Same as WPO = -1. plus omit body C_p output.
			≤-1. Same as WPO = 0. plus omit Mach chart output.
			≤0. No crude grid output.
			≤1. Crude grid output for diagnostic purposes.
			≥2. Same as WPO = 1. plus boundary layer information.
			≥3.. Same as WPO = 2. plus print C_p values off wing and canard.

CARD NUMBER	CARD COLUMN	VARIABLE NAME	DESCRIPTION
Card 2-A (cont'd)			≥4. Same as WPO = 3. plus print solution convergence information for every span-wise plane.
	41-50	AXIT	Number of initial crude grid iterations.
	51-60	AXITF	Number of crude/fine grid iteration cycles.
	61-70	VISMOD	= 1. No viscous effects.
			= 2. Viscous effects computed at end of inviscid analysis.
			= 3. Inviscid/viscous interaction.
Card 3-A	1-10	FSAVE	= 1. Save the flow solution on Unit 99.
			≠ 1. Do not save the flow solution.
	11-20	FSTRT	= 1. Restart the flow solution from Unit 98.
			≠ 1. Do not restart the flow solution.
NOTE:	See Usage Note 1.		
	21-30	CNVTST	Convergence test based on average flow solution correction (DPMAV). Default is 1.0E-05.
	31-40	FCASM2	= 1. Construct wing-canard grid, but run canard-off solution. Will read wing-canard of wing-body-canard input (CASE = 4. or 5. on Card 2-A), construct grids and then do wing alone or wing-body solution (CASE changed to CASE-2. for the solution).
			≠ 1. Do not "turn off" canard.
	41-50	PCTLE	Crude grid leading edge spacing tolerance. Default is 0.01, minimum allowed is 0.005.
NOTE:	See Usage Note 2.		

CARD NUMBER	CARD COLUMN	VARIABLE NAME	DESCRIPTION
Card 3-A (cont'd)	51-60	FCGRD	= 1. Places the canard surface in the vertical mid-location of the canard fine grid system. Appropriate for lightly or negatively loaded canards.
			≠1. Places the canard surface at the lower quarter vertical location in the canard fine grid system.
	61-70	FMESH	= 1. Changes crude grid dimensions to 51,21,27 X,Y,Z points, respectively.
			≠ 1. Default crude grid dimensions of 51,26,31 X,Y,Z points, respectively.
Card 4-A	1-10	SREF	Reference area, if $\leq 0.$, code will calculate.
	11-20	AMAC	Mean aerodynamic chord, if $\leq 0.$, code will calculate.
	21-30	ALAM	Reference taper ratio, if $\leq 0.$, code will calculate.
	31-40	XMOM	X-position for pitching moment reference.
	41-50	ZMOM	Z-position for pitching moment reference.
	51-60	RE	Reynolds Number $\times 10^{-6}$, based on AMAC.
	61-70	FYINT	= 0. Nondimensional ordinate spanwise interpolation.
			= 1. Physical ordinate spanwise interpolation.

NOTE: See Usage Note 3.

NOTE: Omit Card 1-C for CASE<4.

CARD NUMBER	CARD COLUMN	VARIABLE NAME	DESCRIPTION
Card 1-C	1-10	ASECT	Number of streamwise sections defining canard planform ($2. \leq ASECT \leq 20.$).
			ASECT for canard plus ASECT for wing must not exceed 20.
	11-20	AMIN	Number of ordinates defining each canard section ($ANIN \leq 60$).

CARD NUMBER	CARD COLUMN	VARIABLE NAME	DESCRIPTION
Card 1-C (cont'd)	21-30	ANOSW	= 0. Sharp nose canard section. = 1. Blunt nose canard section.
	31-40	ZWINGC	Z-position of canard (waterline).
	41-50	XTRNC	Transition location for canard, streamwise. > 0. fraction of chord < 0. physical distance from leading edge. = 0. default to fixed chord fraction of 0.05.
NOTE:	See Usage Note 4.		
	51-60	CINSDS	Canard incidence, degrees.
	61-70	SHIFTC	Grid shift for CASE>3.
NOTE:	See Usage Note 5.		
Card 1-W	1-10	ASECT	Number of streamwise sections planform ($2. \le ASECT \le 20.$). ASECT for canard plus ASECT for wing must not exceed 20.
	11-20	ANIN	Number of ordinates defining each wing section ($ANIN \le 60.$).
	21-30	ANOSW	= 0. Sharp nose wing sections. = 1. Blunt nose wing sections.
	31-40	ZWING	Z-position of wing (waterline).
	41-50	XTRNW	Transition location for wing, streamwise. Same usage as for XTRNC (See Card 1-C and Usage Note 4.).
	51-60	WINSDS	Wing incidence, degrees.
	61-70	SHIFTW	Grid shift for CASE<4.
NOTE:	See Usage Note 5.		

NOTE: Omit Card Set 2-C through 5-C for CASE<4. Card Set 2-C through 5-C is repeated ASECT (for canard) times.

Card 2-C	1-10	XPL	Canard section leading edge (X-value).
	11-20	YP	Canard section span position (Y-value). First Y-value must be 0.0 (symmetry plane), even for wing-body case.
	21-30	XPT	Wing section trailing edge (X-value).
	31-40	TWIST	Wing section local incidence (twist angle in degrees).
	41-50	AKODE	= 0. Section ordinates identical to preceding section (omit cards 3-C through 5-C).
			= 1. New section definition expected on cards 4-C and 5-C.
Card 3-C	1-70	XINWC	Canard section x/c coordinates (cards 3-C defined only for first canard section, ANIN values).
Card 4-C	1-70	YINU	Canard section upper surface y/c coordinates (ANIN values).
Card 5-C	1-70	YINL	Canard section lower surface y/c coordinates (ANIN values).

NOTE: Card Set 2-W through 5-W is repeated ASECT (for wing) times.

Card 2-W	1-10	XPL	Wing section leading edge (X-value).
	11-20	YP	Wing section span position (Y-value). First Y-value must be 0.0 (symmetry plane), even for wing-body case.
	21-30	XPT	Wing section trailing edge (X-value).
	31-40	TWIST	Wing section local incidence (twist angle in degrees).
	41-50	AKODE	= 0. Section ordinates identical to preceding section (omit cards 3-W through 5-W)
			= 1. New section definition expected on cards 4-W and 5-W.

CARD NUMBER	CARD COLUMN	VARIABLE NAME	DESCRIPTION
Card 3-W	1-70	XINW	Wing section x/c coordinates (cards 3-W defined only for first wing section, ANIN values expected).
Card 4-W	1-70	YINU	Wing section upper surface y/c coordinates (ANIN values).
Card 5-W	1-70	YINL	Wing section lower surface y/c coordinates (ANIN values).

NOTE: Omit card set 1-B through 13-B for CASE = 2 or CASE = 4.

CARD NUMBER	CARD COLUMN	VARIABLE NAME	DESCRIPTION
Card 1-B	1-10	BKOD	= 1. Infinite cylinder (only RADIUS need be input).
			=-1. Same as BKOD = 1. No embedded body grid. Crude grid body representation only.
			= 2. Simple axisymmetric body definition requested (input XBIN, RIN on card(s) 2-B and 3-B.
			=-2. Same as BKOD = 2. No embedded body grid. Crude grid body representation only.
			= 3. Complex body definition requested (input Quick-Geometry model on card(s) 4-B through 13-B).
			=-3. Same as BKOD = 3. No embedded body grid. Crude grid body representation only.

NOTE: BKOD>0. is available for body input checkout only. Flow solution is available for BKOD<0. only.

NOTE: See Usage Note 6.

CARD NUMBER	CARD COLUMN	VARIABLE NAME	DESCRIPTION
	11-20	BNOSE	Body nose (X-value) for BKOD=±2. or ±3.
	21-30	BTAIL	Body tail (X-value) for BKOD=±2. or ±3.
	31-40	BNIN	Number of axisymmetric body coordinates to be input. BNIN $\leq$ 60. (for BKOD = ±2. only).
	41-50	RADIUS	Cylinder radius for BKOD = ±1. only.

CARD NUMBER	CARD COLUMN	VARIABLE NAME	DESCRIPTION
Card 1-B (cont'd)	51-60	ANOSB	= 0. Sharp nose body = 1. Blunt nose body BKOD = ±2. only.

NOTE: Omit card sets 2-B and 3-B for BKOD = ±1. or BKOD = ±3.

CARD NUMBER	CARD COLUMN	VARIABLE NAME	DESCRIPTION
Card(s) 2-B	1-70	XINB	Axisymmetric body X-coordinates (BNIN values).
Card(s) 3-B	1-70	RIN	Axisymmetric body radii (BNIN values).

NOTE: Omit card sets 4-B through 13-B for BKOD = ±1. or BKOD = ±2.

CARD NUMBER	CARD COLUMN	VARIABLE NAME	DESCRIPTION
Card 4-B	1-70	VTITLE	Quick-Geometry model title
Card 5-B	1-10	ACSM	Number of distinct cross-section models (ACSM card sets 6-B and 7-B will follow).
Card 6-B	1-10	ADUM	Running count of current cross-section model (1-ACSM).
	11-20	AARC	Number of arcs in current cross-section model (AARC Card(s) 7-B will follow).
	21-60	CTITLE	Title or descriptor of current cross-section model.
Card 7-B	1-8	ARCNAM	Arc or component name.
	11-14	ASHAPE	Arc or component shape.
	21-28	PNTNAM(1)	Control point name for beginning of this arc.
	32-28	PNTNAM(2)	Control point name for termination of this arc.
	41-48	PNTNAM(3)	Slope control point name for this arc, if required.
Card 8-B	1-10	ANTCSM	Number of cross-section models to define entire body (ANTCSM card(s) 9-B will follow).

CARD NUMBER	CARD COLUMN	VARIABLE NAME	DESCRIPTION
Card 9-B	1-10	ADUM	Running count of current cross-section model (1-ANTCSM).
	11-20	AMODEL	Index corresponding to already defined cross-section models (between 1 and ACSM).
	21-30	XCSMS1	Starting X-station for current cross-section model.
	31-40	XCSMS2	Ending X-station for current cross-section model.
Card 10-B	1-10	BLINE	Number of body line models to be defined by segments (BLINE card set pairs 11-B and 12-B follow).
	11-20	ALIAS	Number of body line models to be aliased (Input ALIAS card(s) 13-B below).
Card 11-B	1-10	BLSEG	Number of segment(s) defining body line model.
	11	BYORZ	The letter Y or Z indicates which data definition is to follow.
	12-19	BNAME	Body line name to be defined.
Card 12-B	1-4	SSHAPE	Segment shape.
	11-20	D(1)	X-station for beginning of segment.
	21-30	D(2)	Y or Z value corresponding to D(1).
	31-40	D(3)	X-station for termination of segment.
	41-50	D(4)	Y or Z value corresponding to D(3).
	51-60	D(5)	X-station for segment slope control point.
	61-70	D(6)	Y or Z value corresponding to D(5).

CARD NUMBER	CARD COLUMN	VARIABLE NAME	DESCRIPTION
Card 13-B	11	BYORZ	The letter Y or Z indicates which data definition is to follow.
	12-19	BNAME	Body line name to be defined.
	21	AYORZ	The letter Y or Z indicates which definition is to be used for aliasing.
	22-29	ANAME	Body line name to which BNAME is aliased.

USAGE NOTES

1. Saved Solution

When an analysis consists of both crude only and then crude/fine iterations, a saved solution is written on unit 99 immediately before the crude/fine iterations begin. At the end of the crude/fine iterations, the previous saved solution is overwritten with the most recent results. In this way, an abnormal termination of the crude/fine iterations will have the crude iteration results saved on unit 99. With an abnormal termination, appropriate job control cards may be needed to "permanently" save the unit 99 data.

2. PCTLE Parameter

The code will shift the input geometry to find a streamwise location in the crude grid where the first points at each span station on the wing and canard is not "too close" to the leading edge. (Riegel's rule is not used for the boundary conditions.) The PCTLE parameter is the required minimum distance from the leading edge. See also Usage Note 5.

3. FYINT Parameter

The input wing and canard section ordinates (YINU AND YINL) are linearly interpolated from the input span stations to the analysis span stations. The method of interpolation is controlled by the parameter FYINT. If FYINT = 0. nondimensional spanwise interpolation of the ordinates is used:

$$(z/c)_Y = (z/c)_{Y1}(1.-R) + (z/c)_{Y2}(R)$$

If FYINT = 1. physical spanwise interpolation of the ordinates is used:

$$(z/c)_Y = (z/c)_{Y1}(1.-R)(c_{Y1}/c_Y) + (z/c)_{Y2}(R)(c_{Y2}/c_Y)$$

where

c = local chord	Y = interpolated span station
R = (Y-Y1)/(Y2-Y1)	Y1,Y2 = input span stations
z/c = nondimensional section ordinate	

The first formula is the more usual analysis code interpolation, while the second formula corresponds to manufacturing lofting methods. Waggoner [10] has shown the difference to be significant for his application. The difference between the two formulas becomes greater for more highly tapered wings and fewer input span stations.

4. Boundary Layer Transition

The fraction of chord designation results in transition specified at the same percentage chord location at all span stations. Thus, the physical distance from the leading edge decreases as the local chord length decreases. The physical distance designation results in transition specified at the same fixed distance from the leading edge. Thus, the percentage chord from the leading edge increases as the local chord length decreases.

5. Grid Shift Parameters

Variables SHIFTC and SHIFTW can be used to set the initial streamwise placement of the configuration. (The grid is fixed in location and the input geometry is shifted.) The code will determinate the appropriate shift with a search procedure. After the first analysis run, the resulting shift parameter can be input to eliminate the search in subsequent runs. The shift parameter can also be used to evaluate different grid placements. If the grid placement search should fail, the shift parameter can be used to start the search at different locations, possibly finding a satisfactory grid placement.

6. Body/Fuselage Geometry Model

The present method allows complex three-dimensional geometries to be input, processed and converted into a suitable array of boundary conditions for analysis. Although the input or modeling of complex body shapes is extremely error prone and certain applications might not warrant this level of effort, it is necessary in aircraft applications when fuselage contours (e.g., canopies, fairings) are required. The code does not provide an embedded fine grid body analysis. The embedded body grid option is provided to allow a detailed body input definition checkout. This will be most useful if the graphical output is available. The following discussion is excerpted from Reference [1]. Additional information and examples will be found in the reference.

The complex fuselage modeling system has been named "Quick-Geometry" by its developers, Vachris and Yaeger [6]. A detailed User's Guide for the Quick-Geometry system can be found in the appendix of Reference [7]. This system was originally developed for the geometric modeling of wing-body shapes. Since only fuselage shapes are of concern here, many of the more sophisticated options including fillets and patches will not be described in the paragraphs which follow. In addition, if References [6] and [7] are being used to augment the modeling description provided herein, it should be noted that the input format has been modified to be more consistent with that of the basic transonic wing-body code.

The geometry package requires that certain body lines and cross-section lines be defined. The body lines and cross-section lines may be likened to the stringers and bulkheads, respectively, used in fuselage construction. These line models are defined by a combination of simple curves (i.e., lines, ellipses, cubics). They are taken together to provide a continuous analytical model of the surface geometry. Slopes and normals are developed analytically. Either discontinuous intersections or smooth fairings can be modeled and enforced.

Two different coordinate systems are employed. Geometry definition is performed in a Cartesian coordinate system (x, y, z), while interrogation of the model for body boundary conditions is performed in cylindrical coordinates (X, R, θ). This results in the use of a plane of symmetry map axis, the height of which usually corresponds to the position of the max-half-breadth line. It is required that the configuration radius at any cross-sectional cut be a single valued function of the angle θ. These definition lines and coordinate systems are illustrated in Figure 6.

A minimum of four body lines are required for the simplest fuselage. These are: 1) top centerline, 2) bottom centerline, 3) max-half-breadth line, and 4) the map axis. Each body line must be defined by both its Y and Z values over the full range of X (between fuselage nose and tail). Similarly, a minimum of two cross-section line segments are required for each different cross-section line model. These are 1) body upper, and 2) body lower.

Both body lines and cross-section lines are specified by defining key arc or segment shapes and their accompanying limiters. The segment shape boundary conditions that are used to determine the coefficients of the slope equation are the 1) origin point, 2) termination point, and 3) slope control point. The slope control point lies at the intersection of the line which is tangent to the segment shape at the origin point and the line which is tangent to the segment shape at the termination point (see Figure 7). The slope control point is a very convenient way of specifying slope conditions. In particular, it allows for the simultaneous specification of slope conditions at both ends of the segment.

Figure 7 is a schematic illustrating the component build-up of a particular body line and cross-section line model. Naturally, LINE segments do not require a slope control point. In this case, the portion of the body top center line illustrated requires four body line segments and the cross-section is constructed with two arcs (two is the minimum number allowed).

The arc shapes used for defining a cross-section line model are listed in Table 1.

TABLE 1. CROSS-SECTION ARC SHAPES

SHAPE	KEYWORD	EQUATION
Line	LINE	$Ay + Bz + C = 0$
Ellipse (Concave to origin)	ELLI	$\frac{(y - y_0)^2}{A^2} + \frac{(z - z_0)^2}{B^2} = 0$
Ellipse (Convex to origin)	ELLO	Same as ELLI.

They are input in an order which starts at the body bottom centerline and proceeds to the body top center line. The segment shapes used for defining a body line model are listed in Table 2.

TABLE 2. BODY LINE SEGMENTS

SHAPE	KEYWORD	EQUATION
Line	LINE	$Ax + By = 0$
X-PARABOLA	XPAR	$Ax + By + y^2 = 0$
Y-PARABOLA	YPAR	$Ax + By + x^2 = 0$
X-ELLIPSE	ELLX	$Ax + By + Cx^2 + y^2 = 0$
Y-ELLIPSE	ELLY	$Ax + By + Cy^2 + x^2 = 0$
CUBIC	CUBI	$Ax + By + Cx^2 + x^3 = 0$

Cross-sections arcs are input in their order of appearance. However, body line segments are defined along with an index which establishes their order in the X-direction. In addition, body lines may be aliased to other body lines. This allows two body line models to have identical mathematical representations without repeating the body line segment input. For example, the Z-value of the map axis (ZMAP) is typically the Z-value of the maximum half-breadth (ZMHB). The two are made identical by aliasing the two body line model names ZMAP and ZMHB.

It should be noted that cross-sections are defined only in terms of named component arcs (arc shape table) and named control points. On the other hand, body lines are defined mathematically by coordinates over the length of the configuration for which they are required. At a given X-station, the body lines are interrogated to give the key control points required to construct the cross-sectional arcs.

SECTION V

CODE OPERATION

The program reads data from unit 5 and writes output on unit 6. Units 20 and 40 are used as scratch files in the COPES routines. The scratch file numbers may be changed by changing two cards at the beginning of the COPES program. The analysis code stores data on unit numbers 1,7,8,15,85,98 and 99. Unit 15 is used by the Ames plotting software for graphical output. It has been left available to facilitate incorporation of graphic capability by other users.

The computer code is written in FORTRAN, employing the CDC overlay structure. Storage requirements on a CDC 7600 are 151000_8 small core memory and 770000_8 large core memory. Typical CPU times on the 7600 for a wing-body-canard analysis with viscous effects is eleven minutes. The CPU time for an optimization run will vary according to the complexity of the optimization problem, as discussed in Section 3.

The overlay structure is described in Table 3. The overlay sizes are shown in Table 4. The job control language (JCL) for operation on the Ames CDC 7600 is shown in Figure 8. The job card and account card will vary with the individual user. Compilation of the entire program requires one minutes of CPU time. To save CPU time for short test runs, a binary file of the code is stored as a permanent file. Thus, the JCL uses an UPDATE,Q command followed by FTN and COPYL to change, compile and replace only those routines being modified for a particular run. The numerical optimization will require changes to the subroutines OGEOM, MOD and ANALIZ (all in Overlay (0,0)). For simple analysis, no changes need be made, except perhaps to BLOCK DATA to change some defaults. The UPDATE,Q always requires some input. The *COMPILE INTEG card in Figure 8 is shown as an example that satisfies this requirement.

TABLE 3. OVERLAY STRUCTURE

OVERLAY	CONTENTS
(0,0)	COPES control program, subroutines needed throughout the code
(1,0)	Input control program, QUICK geometry look-up routines
(1,1)	Input processing
(1,2)	Input geometry plotting, body area calculations
(2,0)	Transonic analysis control program, viscous calculation routines
(2,1)	Potential flow solution routines
(2,2)	Pressure, force and moment calculation and output
(2,3)	Output plotting program
(3,0)	Previously used for opening and closing plotting files. Currently not used.
(4,0)	Body force and moment calculation
(5,0)	CONMIN optimization routines

TABLE 4. OVERLAY SIZES

OVERLAY	SMALL CORE SIZE	LARGE CORE SIZE
(0,0)	66506	766710
(1,0)	101300	766710
(1,1)	125321	767010
(1,2)	104642	766710
(2,0)	105477	766710
(2,1)	137421	766710
(2,2)	136514	766710
(2,3)	112151	766710
(3,0)	66517	766710
(4,0)	101261	766710
(5,0)	100524	766710

The Quick-Geometry routines are needed for the input geometry processing and the body force and moment calculations. The small core restrictions do not allow all the routines to be in Overlay(0,0). The COPYBR cards are used to copy the necessary routines into the overlays where needed (Overlays (1,0), (2,2) and (4,0)).

Not shown in the JCL are the cards for permanently saving the flow solution. The Applied Computational Aerodynamics Branch at Ames can provide the necessary authorization and information.

If the code is converted for use on IBM type computers, several Quick-Geometry variables need special treatment. These variables and the sub-routines where they appear are listed in Table 5. The REAL*8 declaration is needed to have sufficient word length for line model labeling and increase the accuracy of the body model calculations.

TABLE 5. REAL*8 VARIABLES FOR IBM USAGE OF QUICK-GEOMETRY

SUBROUTINE NAME	VARIABLE NAME
CURVES	A, B, C, Y, T, X, FACT, RFACT, S
MODTV	SUM, ONE
VDOTZ	C, ONE
QWIKDE	CPNTNM, COMPNM
BLMCHK	CPNTNM, COMPNM, ANAME, BNAME, BLMNAM, EQUNB, EQUNA
BLMDEF	CPNTNM, COMPNM, ANAME, BNAME, BLMNAM, BLANK2
CSMCHK	CPNTNM, COMPNM, ARCNAM, PNTNAM, BLANK2, ZMAPNM, ARCNM
CSMDEF	CPNTNM, COMPNM, ARCNAM, PNTNAM, BLANK2, ZMAPNM, ARCNM, AMAPAX
GEMOUT	CPNTNM, COMPNM, ARCNM, PNTNAM, BLANK2, AMAPNM, ARCNM
DSETUP	ALABLE, BLABLE

The program output listing of the input data for a sample analysis run is shown in Figure 9. In front of this input were the COPES control cards:

```
TITLE
1,
END
```

with no delimiters in between. The input example uses NACA0010 wing sections. Following the input data listing is a data read echo showing the information the program has read from the input cards (Figure 10). Portions of the grid generation output are shown in Figures 11 and 12.

In this example, the flowfield solution begins with five crude grid iterations, and then continues with five fine grid/crude grid iterations. The program output for this is shown in Figure 13. When the flow solution is completed, detailed flow information is output at each analysis station on the wing and canard. An example for station 11 is shown in Figure 14.

Force and moment output for the wing, canard and body is shown in Figure 15. Figure 16 shows the final wing-body-canard spanload output. All of the results shown in Figure 10 through 16 were for the analysis input data of Figure 9.

APPENDIX - GLOSSARY OF OUTPUT VARIABLES

AGn	n=1,2,3,4,5 Coefficients a_n in equation (7).
A1	Coefficient A_1 in equation (6).
A2	Coefficient A_2 in equation (8).
A3	Coefficient A_3 in equation (9).
CCD/CAV	Spanload drag.
CCF/CAV	Spanload skin friction.
CCL/CAV	Spanload lift.
CCM/CAV/MAC	Spanload pitching moment.
CD	Drag coefficient (local or total, as appropriate).
CDINT	Local drag coefficient in spanload tables.
CF	Skin friction coefficient.
CIR	Circulation.
CL	Lift coefficient (local or total as appropriate).
CLINT	Local lift coefficient in spanload tables.
CM	Pitching moment coefficient.
CMLOC	Local pitching moment coefficient (about quarter chord point).
CP	Pressure coefficient.
C1	Coefficient C_1 in equation (8).
C3	Coefficient C_3 in equation (8).
DPM	Maximum correction to flow potential for the current iteration.
DPMAV	Average of the absolute value of all the flowfield corrections for the current iteration.
DRAG	Local drag contribution in the body force output.
D1	Coefficient D_1 in equation (9).
D3	Coefficient D_3 in equation (9).

ETA	Spanwise computational ordinate.
I	General streamwise index.
IL	Streamwise index of leading edge point in crude grid.
INOSE	Streamwise index of the first point on the body in the body fine grid.
INOSEC	Streamwise index of the first point on the body in the crude grid.
IT	Streamwise index of the trailing edge point at an analysis station in the crude grid.
ITAIL	Streamwise index of the last point on the body in the body fine grid.
ITAILC	Streamwise index of the last point on the body in the crude grid.
ITER	Iteration count.
J	General spanwise index.
JSD	Spanwise index of first solution plane outside the body computational surface.
KLO	Vertical index for bottom of body computational surface.
LOAD	Local lift contribution in body force output.
NSP	Number of supersonic points in the crude grid.
PCT	Fractional cord distance from leading or trailing edge during crude grid shift search.
PHI	Pertubation velocity potential.
RMAX	Maximum body radius.
RSDAV	Average of the absolute value of the error in the finitie difference equation at all grid points for the current iteration.
SEXPC	Canard exposed area.
SHIFT	Grid shift parameter.
U	Streamwise pertubation velocity.
V	Spanwise pertubation velocity.
WCORD	Local cord for wing or canard.
X	Streamwise physical coordinate.
XI	Streamwise computational coordinate.

XLE	Wing or canard leading edge X-location.
XNOSE	Body nose X-location.
XTAIL	Body tail X-location.
XTE	Wing or canard trailing edge location.
XWF	Wing or canard fine grid streamwise physical coordinate.
X/C	Percent fraction of local cord.
Y	Spanwise physical coordinate.
Y/C	Percent chord wing or canard section ordinate.
Z	Vertical physical coordinate.
ZETA	Vertical computational coordinate.
2Y/B	Fraction of semispan (on wing or canard as appropriate).

REFERENCES

1. Boppe, C. W.: "Transonic Flowfield Analysis for Wing-Fuselage Combinations," NASA Contractor Report 3243, May 1980.

2. Mason, W. H., et al.: "An Automated Procedure for Computing the Three-Dimensional Transonic Flow Over Wing-Body Combinations, including Viscous Effects," Report AFFDL-TR-77-122, Vol. 1, October 1977.

3. Nash, J. F. and Tseng, R. R.: "The Three-Dimensional Turbulent Boundary Layer on an Infinite Yawed Wing," The Aeronautical Quarterly, November 1971.

4. Bradshaw, P. and Ferriss, D. H.: "Calculation of Boundary Layer Development Using the Turbulent Energy Equation. Compressible Flow on Adiabatic Walls," J. Fluid Mech., Vol. 46, 1971.

5. Hinson, B. L. and Burdges, K. P., "Acquisition and Application of Transonic Wing and Far-Field Test Data for Three-Dimensional Computational Method Evaluation," Technical Report, AFOSR-TR-80-0421, March 1980.

6. Vachris, A. F. and Yaeger, L. S.: "Quick-Geometry - A Rapid Response Method for Mathematically Modeling Configuration Geometry," NASA SP-390, October 1975, pp. 49-73.

7. Marconi, F. and Yaeger, L.: "Development of A Computer Code for Calculating the Steady Super/Hypersonic Inviscid Flow Around Real Configurations," NASA CR-2626 (Vol. II), May 1976.

8. Vanderplaats, G. N., "COPES--Control Program for Engineering Synthesis," to be published as a Naval Postgraduate School Memorandum.

9. Vanderplaats, G. N., "CONMIN--A Fortran Program for Constrained Function Minimization," NASA TM X-62282, August 1973.

10. Waggoner, E. G.: "Computational Transonic Analysis for a Supercritical Transport Wing-Body Configurations," AIAA Paper 80-129, January 1980.

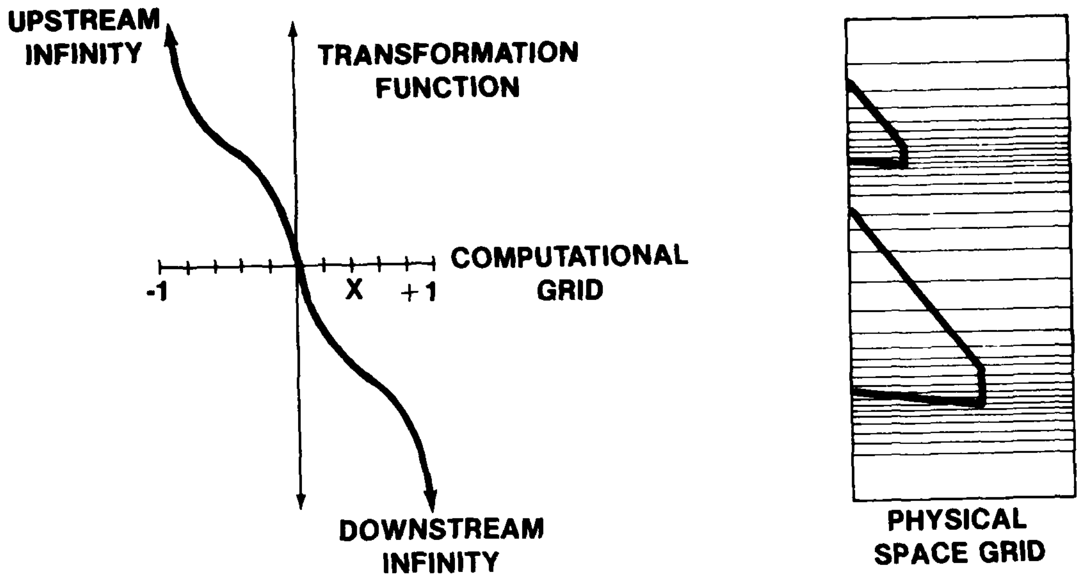


Figure 1. Wing-canard crude mesh transformation

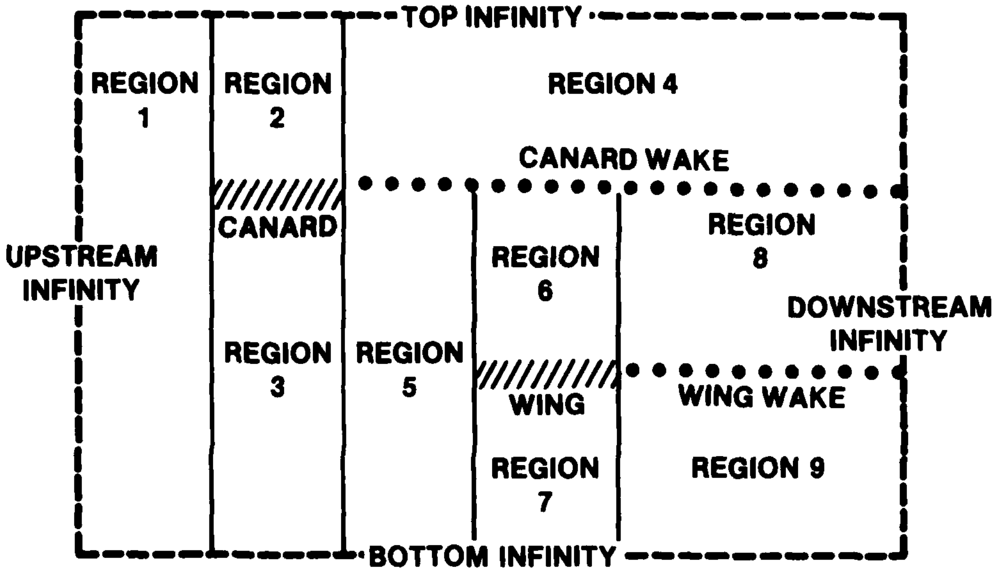


Figure 2. Multiple wake vertical line relaxation regions, crude grid

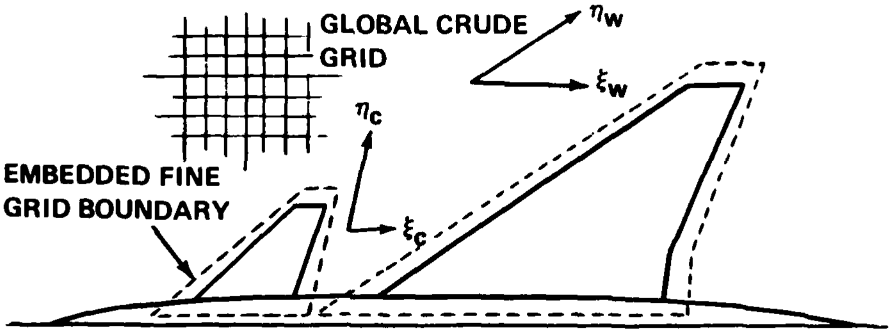


Figure 3. Embedded fine grid system

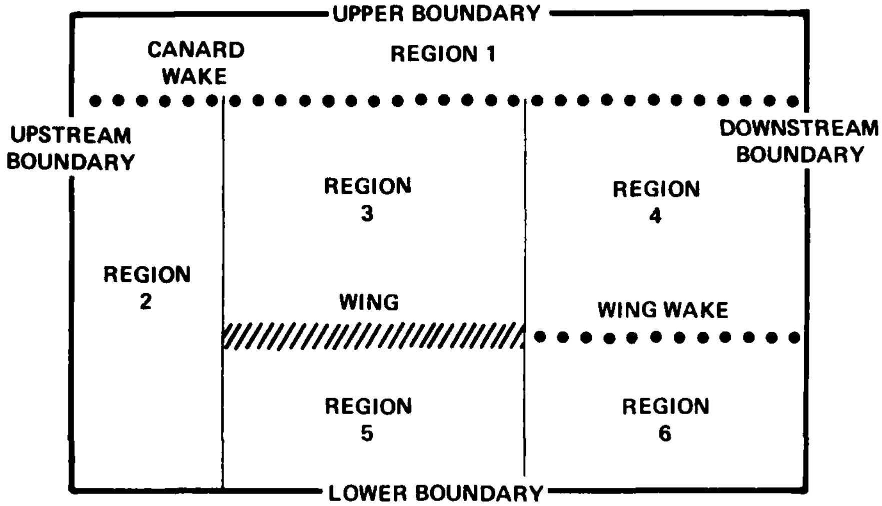


Figure 4. Multiple wake vertical line relaxation regions, fine grid

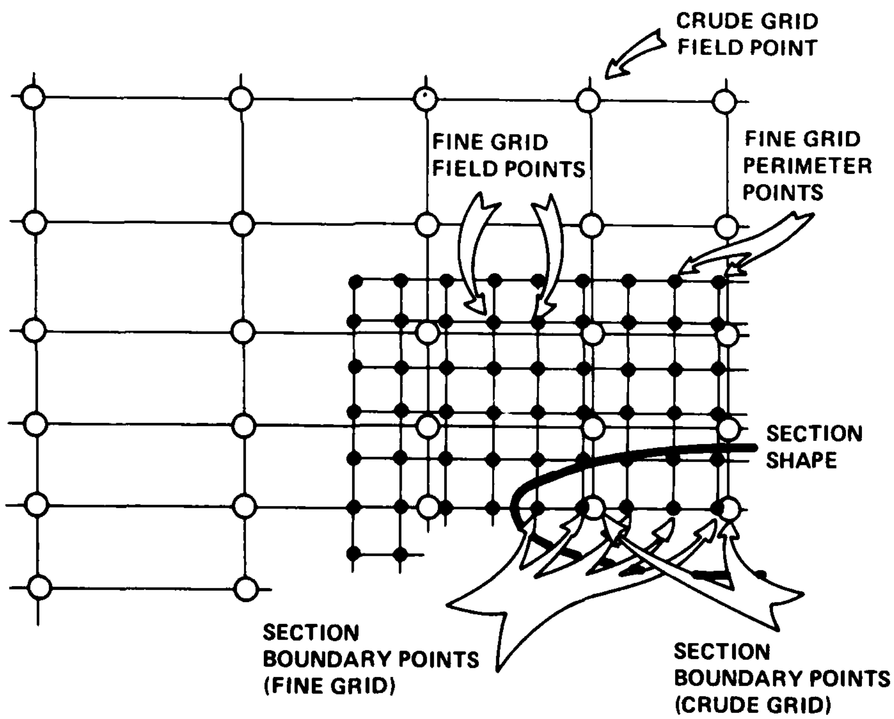


Figure 5. Fine/crude grid interface (from [1])

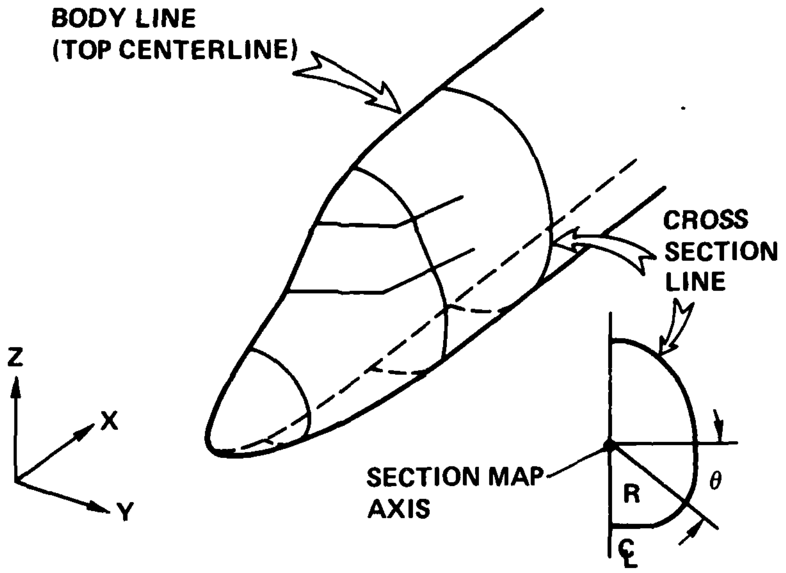


Figure 6. Quick geometry model lines and coordinate system (from [1])

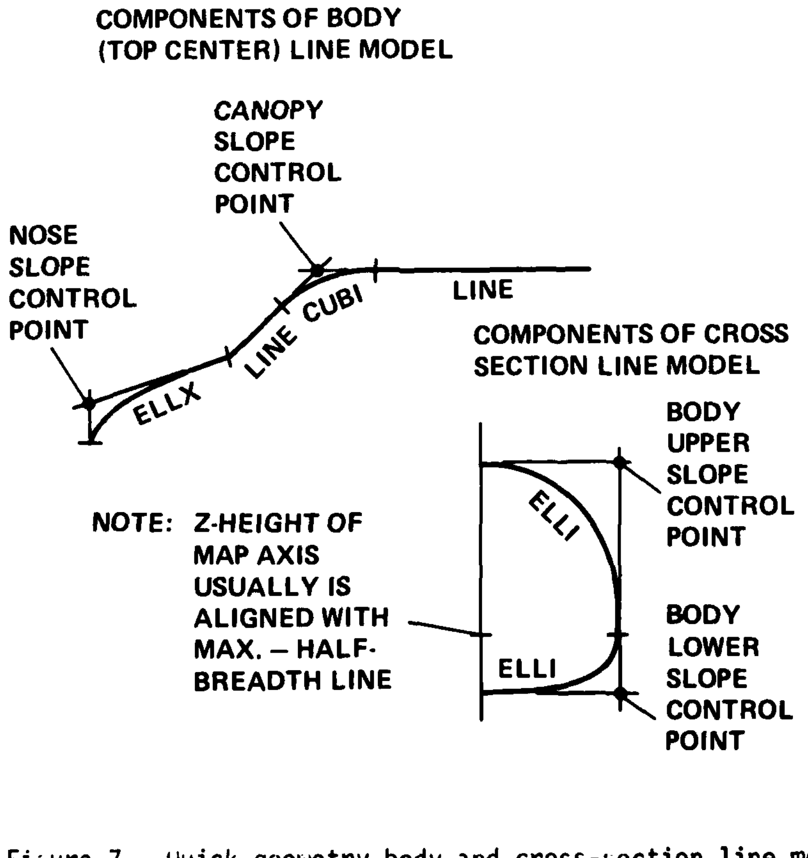


Figure 7. Quick geometry body and cross-section line models (from [1])

```
--------- JOB CARD ---------
--------- ACCOUNT CARD ---------
ATTACH,OLDPL,PANDORAS,ID=STAGRUM.
UPDATE,Q.
RETURN,OLDPL.
RFL,160000
FTN,L=0,I=COMPILE,B=LGOA,LCM=I,OPT=2.
REDUCE.
RETURN,COMPILE.
ATTACH,LGOLD,PANDORAB,ID=STAGRUM.
REWIND,LGOLD,LGOA.
COPYL,LGOLD,LGOA,LGO1,,R.
RETURN,LGOLD.
RETURN,LGOA.
REWIND,LGO1.
COPYBR,LGO1,LGO2,52.
COPYBR,LGO1,TEMP,21.
REWIND,TEMP.
COPYBR,TEMP,LGO2,21.
REWIND,TEMP.
COPYBR,LGO1,LGO2,56.
COPYBR,LGO1,TEMP2,1.
REWIND,TEMP2.
COPYBR,TEMP2,LGO2,1.
REWIND,TEMP2.
COPYBR,TEMP,LGO2,21.
REWIND,TEMP.
COPYBR,LGO1,LGO2,14.
COPYBR,TEMP2,LGO2,1.
COPYBR,TEMP,LGO2,21.
COPYBR,LGO1,LGO2,11.
RETURN,LGO1.
RETURN,TEMP.
RETURN,TEMP2.
LDSET(PRESET=INDEF)
LGO2(PL=200000)
7/8/9 CARD DELIMITER
*IDENT DUMMY
*COMPILE INTEG
--------- ADDITIONAL UPDATE IDENTS HERE ---------
7/8/9 CARD DELIMITER
--------- COPES DATA BEGINS HERE ---------
--------- FOLLOWED BY FLOW ANALYSIS DATA ---------
6/7/8/9/ CARD DELIMITER
```

Figure 8 - Job control language

```
PANDORA---PRELIMINARY AUTOMATED NUMERICAL DESIGN OF REALISTIC AIRCRAFT

DEVELOPED AT GRUMMAN AEROSPACE CORPORATION,BETHPAGE, NEW YORK
FOR AIR FORCE FLIGHT DYNAMICS LABORATORY, WPAFB, OHIO
CAPTAIN ROBERT A. LARGE, CONTRACT MONITOR (513-255-5564)


INPUT DATA LISTING


          ++++++++++++++++++++++++++++++++

FORWARD SWEPT SAMPLE CASE
5.0       0.8       1.0       1.0       5.0       5.0       1.0

0.0
3.0       18.       1.0       0.0       .05       -2.0
5.0       18.       1.0       -30.      .05       1.0
334.      0.0       423.      0.0       1.0
0.00      1.25      2.50      5.00      7.50      10.0      15.0
20.0      25.0      30.0      40.0      50.0      60.0      70.0
80.0      90.0      95.0      100.
0.000     1.578     2.178     2.962     3.500     3.902     4.455
4.782     4.952     5.002     4.837     4.412     3.803     3.053
2.187     1.207     0.672     0.105
-0.000    -1.578    -2.178    -2.962    -3.500    -3.902    -4.455
-4.782    -4.952    -5.002    -4.837    -4.412    -3.803    -3.053
-2.187    -1.207    -0.672    -0.105
334.      37.       423.      0.0       0.
373.      80.0      403.      0.0       0.
444.      0.        584.5     0.0       1.0
0.00      1.25      2.50      5.00      7.50      10.0      15.0
20.0      25.0      30.0      40.0      50.0      60.0      70.0
80.0      90.0      95.0      100.
0.000     1.578     2.178     2.962     3.500     3.902     4.455
4.782     4.952     5.002     4.837     4.412     3.803     3.053
2.187     1.207     0.672     0.105
-0.000    -1.578    -2.178    -2.962    -3.500    -3.902    -4.455
-4.782    -4.952    -5.002    -4.837    -4.412    -3.803    -3.053
-2.187    -1.207    -0.672    -0.105
444.      37.       584.5     0.0       0.0
462.5     64.00     552.0     0.0       0.
435.2     114.0     503.5     0.0       0.
408.0     164.0     454.5     0.0       0.
-2.0      100.      700.      11.0      37.0      0.0
100.      160.      220.      280.      340.      400.      460.
520.      580.      640.      700.
0.0       13.32     23.68     31.08     35.52     37.00     35.52
31.08     23.68     13.32     0.0
          ++++++++++++++++++++++++++++++++
```

Figure 9. Analysis code input data listing

```
DATA READ ECHO     (DEFAULTS AND INTEGER CONVERSION INCLUDED)
INPUT CARD 1-A     FORWARD SWEPT SAMPLE CASE
INPUT CARD 2-A     CASE=  5  AMACH= .800  AOA= 1.000  WPO=  1  AXIT=   5  AXITF=   5  VISMOD=  1
INPUT CARD 3-A     FSAVE=  0  FSTRT=  0  CNVTST= 1.000E-05  NCASM2=  0  PCTLE=     .010  FCGRD=  0  FMESH=  0
INPUT CARD 4-A     SREF=       0.000  AMAC=   0.000  ALAM= 0.000      (PROGRAM WILL CALCULATE VALUES IF ZERO)
                    XMOM=   0.000  ZMOM=   0.000  RE=   1.000E+07  FYINT=  0
INPUT CARD 1-C     ASECT=  3  ANIN= 18  ANOSW=  1  ZWINGC=      0.000  XTRNC=    .050  CINSDS=  -2.000  SHIFTC=      0.000
INPUT CARD 1-W     ASECT=  5  ANIN= 18  ANOSW=  1  ZWING=   -30.000  XTRNW=    .050  WINSDS=    1.000  SHIFTW=      0.000
INPUT CARD  1  OF CARD SET 2-C     XPL=   334.000  YP=     0.000  XPT=    423.000  TWIST= 0.000  AKODE=  1
INPUT CARD  2  OF CARD SET 2-C     XPL=   334.000  YP=    37.000  XPT=    423.000  TWIST= 0.000  AKODE=  0
INPUT CARD  3  OF CARD SET 2-C     XPL=   373.000  YP=    80.000  XPT=    403.000  TWIST= 0.000  AKODE=  0
INPUT CARD  1  OF CARD SET 2-W     XPL=   444.000  YP=     0.000  XPT=    584.500  TWIST= 0.000  AKODE=  1
INPUT CARD  2  OF CARD SET 2-W     XPL=   444.000  YP=    37.000  XPT=    584.500  TWIST= 0.000  AKODE=  0
INPUT CARD  3  OF CARD SET 2-W     XPL=   462.500  YP=    64.000  XPT=    552.000  TWIST= 0.000  AKODE=  0
INPUT CARD  4  OF CARD SET 2-W     XPL=   435.200  YP=   114.000  XPT=    503.500  TWIST= 0.000  AKODE=  0
INPUT CARD  5  OF CARD SET 2-W     XPL=   408.000  YP=   164.000  XPT=    454.500  TWIST= 0.000  AKODE=  0
INPUT CARD 1-B     BKOD= -2  BNOSE=    100.000  BTAIL=    700.000  BNIN= 11  RADIUS=37.000  ANOSB=  0
```

Figure 10. Analysis code data read echo

CRUDE GRID

GRID PARAMETERS

AG1=-158.3056 AG2= 19.2807 AG3= -38.9485 AG4= -3.2134 AG5= 1.2789 AGS= 775.7780

A2= .00529 C1= -17.3184 C3= 35.3437

A3= 72.00221 D1= -7.7154 D3= 0.0000

51 X GRID POINTS 26 Y GRID POINTS 31 Z GRID POINTS

	COMPUTATIONAL DOMAIN	PHYSICAL DOMAIN
GRID POINT	XI	X
1	-3.16443	UPSTREAM INFINITY
2	-3.03785	-8381.77404
3	-2.91127	-3588.21549
4	-2.78470	-2030.73070
5	-2.65812	-1283.76775
6	-2.53154	-861.34814
7	-2.40497	-600.85758
8	-2.27839	-432.15454
9	-2.15181	-319.84357
10	-2.02523	-244.02960
11	-1.89866	-192.61656
12	-1.77208	-157.80386
13	-1.64550	-134.33204
14	-1.51893	-118.53557
15	-1.39235	-107.79939
16	-1.26577	-100.23051
17	-1.13919	-94.45074
18	-1.01262	-89.46052
19	-.88604	-84.54621
20	-.75946	-79.21462
21	-.63289	-73.14543
22	-.50631	-66.15515
23	-.37973	-58.16904
24	-.25315	-49.19833
25	-.12658	-39.32103
26	.00000	-28.66508
27	.12658	-17.39295
28	.25315	-5.68693
29	.37973	6.26565
30	.50631	18.28779
31	.63289	30.22970
32	.75946	41.98793
33	.88604	53.52821
34	1.01262	64.91349
35	1.13919	76.33999
36	1.26577	88.18483
37	1.39235	101.07140
38	1.51893	115.96216
39	1.64550	134.29475
40	1.77208	158.18932
41	1.89866	190.77686
42	2.02523	236.74272
43	2.15181	303.27378
44	2.27839	401.81280

Figure 11. Crude grid generation output

45	2.40497	551.56196
46	2.53154	787.18420
47	2.65812	1178.04896
48	2.78470	1885.95886
49	2.91127	3396.04111
50	3.03785	8132.95675
51	3.16443	DOWNSTREAM INFINITY

Y GRID POINT	ETA	Y
1	0.00000	0.00000
2	.04000	6.87726
3	.08000	13.79242
4	.12000	20.78383
5	.16000	27.89083
6	.20000	35.15427
7	.24000	42.61711
8	.28000	50.32522
9	.32000	58.32822
10	.36000	66.68065
11	.40000	75.44345
12	.44000	84.68590
13	.48000	94.48828
14	.52000	104.94554
15	.56000	116.17260
16	.60000	128.31210
17	.64000	141.54630
18	.68000	156.11589
19	.72000	172.35154
20	.76000	190.72981
21	.80000	211.98056
22	.84000	237.31561
23	.88000	268.99285
24	.92000	312.06181
25	.96000	382.60113
26	1.00000	+ Y INFINITY

Z GRID POINT	ZETA	Z
1	-1.00000	- Z INFINITY
2	-.93333	-677.85416
3	-.86667	-332.05704
4	-.80000	-215.42766
5	-.73333	-156.06162
6	-.66667	-119.56786
7	-.60000	-94.47328
8	-.53333	-75.85166
9	-.46667	-61.23054
10	-.40000	-49.22649
11	-.33333	-38.99868
12	-.26667	-30.00000
13	-.20000	-21.85185
14	-.13333	-14.27582
15	-.06667	-7.05337
16	0.00000	0.00000
17	.06667	7.05337
18	.13333	14.27582
19	.20000	21.85185
20	.26667	30.00000
21	.33333	38.99868
22	.40000	49.22649
23	.46667	61.23054
24	.53333	75.85166

Figure 11. (Continued)

25	.60000	94.47328
26	.66667	119.56786
27	.73333	156.06162
28	.80000	215.42766
29	.86667	332.05704
30	.93333	677.85416
31	1.00000	+ Z INFINITY

Figure 11. (concluded)

FINE GRID

120 X GRID POINTS   19 Z GRID POINTS

PHYSICAL DOMAIN - X GRID POINTS

SPAN STATION 6

-45.25968	-43.49238	-41.72509	-39.95779	-38.19050	-36.42320	-34.65591	-32.88861	-31.12131	-29.35402
-27.58672	-25.81943	-24.05213	-22.28484	-20.51754	-18.75025	-16.98295	-15.21565	-13.44836	-11.68106
-9.91377	-8.14647	-6.37918	-4.61188	-2.84459	-1.07729	.69001	2.45730	4.22460	5.99189
7.75919	9.52648	11.29378	13.06108	14.82837	16.59567	18.36296	20.13026	21.89755	23.66485
25.43214	27.19944	28.96674	30.73403	32.50133	34.26862	36.03592	37.80321	39.57051	41.33780
43.10510	44.87240	46.63969	48.40699	50.17428	51.94158	53.70887	55.47617	57.24346	59.01076
60.77806	62.54535	64.31265	66.07994	67.84724	69.61453	71.38183	73.14913	74.91642	76.68372
78.45101	80.21831	81.98560	83.75290	85.52019	87.28749	89.05479	90.82208	92.58938	94.35667
96.12397	97.89126	99.65856	101.42585	103.19315	104.96045	106.72774	108.49504	110.26233	112.02963
113.79692	115.56422	117.33152	119.09881	120.86611	122.63340	124.40070	126.16799	127.93529	129.70258
131.46988	133.23718	135.00447	136.77177	138.53906	140.30636	142.07365	143.84095	145.60824	147.37554
149.14284	150.91013	152.67743	154.44472	156.21202	157.97931	159.74661	161.51391	163.28120	165.04850

SPAN STATION 7

-39.20882	-37.57499	-35.94115	-34.30732	-32.67348	-31.03964	-29.40581	-27.77197	-26.13814	-24.50430
-22.87047	-21.23663	-19.60280	-17.96896	-16.33513	-14.70129	-13.06746	-11.43362	-9.79979	-8.16595
-6.53212	-4.89828	-3.26445	-1.63061	.00322	1.63706	3.27089	4.90473	6.53856	8.17240
9.80624	11.44007	13.07391	14.70774	16.34158	17.97541	19.60925	21.24308	22.87692	24.51075
26.14459	27.77842	29.41226	31.04609	32.67993	34.31376	35.94760	37.58143	39.21527	40.84910
42.48294	44.11677	45.75061	47.38444	49.01828	50.65212	52.28595	53.91979	55.55362	57.18746
58.82129	60.45513	62.08896	63.72280	65.35663	66.99047	68.62430	70.25814	71.89197	73.52581
75.15964	76.79348	78.42731	80.06115	81.69498	83.32882	84.96265	86.59649	88.23032	89.86416
91.49800	93.13183	94.76567	96.39950	98.03334	99.66717	101.30101	102.93484	10[illegible]56868	106.20251
107.83635	109.47018	111.10402	112.73785	114.37169	116.00552	117.63936	119.27319	12[illegible]90703	122.54086
124.17470	125.80853	127.44237	129.07620	130.71004	132.34388	133.97771	135.61155	137.24538	138.87922
140.51305	142.14689	143.78072	145.41456	147.04839	148.68223	150.31606	151.94990	153.58373	155.21757

SPAN STATION 8

-30.90550	-29.45481	-28.00411	-26.55342	-25.10273	-23.65203	-22.20134	-20.75065	-19.29995	-17.84926
-16.39857	-14.94787	-13.49718	-12.04649	-10.59579	-9.14510	-7.69440	-6.24371	-4.79302	-3.34232
-1.89163	-.44094	1.00976	2.46045	3.91114	5.36184	6.81253	8.26322	9.71392	11.16461
12.61531	14.06600	15.51669	16.96739	18.41808	19.86877	21.31947	22.77016	24.22085	25.67155
27.12224	28.57293	30.02363	31.47432	32.92501	34.37571	35.82640	37.27710	38.72779	40.17848
41.62918	43.07987	44.53056	45.98126	47.43195	48.88264	50.33334	51.78403	53.23472	54.68542
56.13611	57.58680	59.03750	60.48819	61.93889	63.38958	64.84027	66.29097	67.74166	69.19235
70.64305	72.09374	73.54443	74.99513	76.44582	77.89651	79.34721	80.79790	82.24860	83.69929
85.14998	86.60068	88.05137	89.50206	90.95276	92.40345	93.85414	95.30484	96.75553	98.20622
99.65692	101.10761	102.55830	104.00900	105.45969	106.91039	108.36108	109.81177	111.26247	112.71316
114.16385	115.61455	117.06524	118.51593	119.96663	121.41732	122.86801	124.31871	125.76940	127.22009
128.67079	130.12148	131.57218	133.02287	134.47356	135.92426	137.37495	138.82564	140.27634	141.72703

Figure 12. Example of fine grid output

-71.39598	-70.93998	-70.48398	-70.02798	-69.57198	-69.11598	-68.65998	-68.20398	-67.74798	-67.29198
-66.83598	-66.37998	-65.92398	-65.46798	-65.01198	-64.55598	-64.09998	-63.64398	-63.18798	-62.73198
-62.27598	-61.81998	-61.36398	-60.90798	-60.45198	-59.99598	-59.53998	-59.08398	-58.62798	-58.17198
-57.71598	-57.25998	-56.80398	-56.34798	-55.89198	-55.43598	-54.97998	-54.52398	-54.06798	-53.61198
-53.15598	-52.69998	-52.24398	-51.78798	-51.33198	-50.87598	-50.41998	-49.96398	-49.50798	-49.05198
-48.59598	-48.13998	-47.68398	-47.22798	-46.77198	-46.31598	-45.85998	-45.40398	-44.94798	-44.49198

PHYSICAL DOMAIN - Z GRID POINTS

INPUT SHIFTED TO PUT WING PLANE AT Z=0.0

WING

-11.52325	-9.21860	-6.91395	-4.60930	-2.30465	0.00000	2.30465	4.60930	6.91395	9.21860
11.52325	13.82790	16.13255	18.43720	20.74184	23.04649	25.35114	27.65579	29.96044	

CANARD

22.17175	23.73740	25.30305	26.86870	28.43435	30.00000	31.56565	33.13130	34.69695	36.26260
37.82825	39.39390	40.95955	42.52520	44.09085	45.65650	47.22215	48.78780	50.35345	

Figure 12. (Concluded)

```
        10.246INCREMENTAL CPU TIME, SECONDS        10.246TOTAL CPU TIME, SECONDS

CRUDE ITER NO.   1   DELTA PHI MAX =   5.194E-02   I = 576   J =  17   K =  16   NSP =   0
CIR =  0.         0.         0.         0.         0.         -7.062E-01 -6.027E-01 -4.116E-01 -5.024E-01
CIR =  -3.236E-01
CIR =  0.         0.         0.         2.876E-01  2.889E-01  5.970E-01  1.024E+00  5.842E-01  8.986E-01  6.161E-01
CIR =  6.673E-01  7.229E-01  7.807E-01  8.435E-01  9.097E-01  9.717E-01  1.006E+00  9.316E-01

CRUDE ITER NO.   2   DELTA PHI MAX =   1.367E-02   I =  24   J =  17   K =  16   NSP =   2
CIR =  0.         0.         0.         0.         -1.093E-01 -8.164E-01 -8.104E-01 -7.601E-01 -6.045E-01
CIR =  -4.278E-01
CIR =  0.         0.         0.         6.649E-01  2.756E-01  1.955E+00  2.835E+00  2.364E+00  1.978E+00  1.820E+00
CIR =  1.842E+00  1.891E+00  1.941E+00  1.953E+00  1.918E+00  1.805E+00  1.568E+00  1.108E+00

CRUDE ITER NO.   3   DELTA PHI MAX =   9.404E-03   I =  38   J =   8   K =  16   NSP =   3
CIR =  0.         0.         0.         0.         -8.285E-01 -9.451E-01 -8.805E-01 -7.850E-01 -7.363E-01
CIR =  -4.335E-01
CIR =  0.         0.         0.         6.803E-01  8.432E-01  3.822E+00  4.195E+00  3.715E+00  3.089E+00  2.594E+00
CIR =  2.592E+00  2.551E+00  2.521E+00  2.464E+00  2.339E+00  2.167E+00  1.916E+00  1.502E+00

CRUDE ITER NO.   4   DELTA PHI MAX =   6.318E-03   I =  28   J =  11   K =  16   NSP =   3
CIR =  0.         0.         0.         0.         -9.000E-01 -1.038E+00 -1.009E+00 -8.962E-01 -8.262E-01
CIR =  -5.778E-01
CIR =  0.         0.         0.         1.081E+00  8.544E-01  5.079E+00  4.968E+00  4.373E+00  4.025E+00  3.360E+00
CIR =  3.009E+00  2.969E+00  2.938E+00  2.897E+00  2.800E+00  2.573E+00  2.189E+00  1.496E+00

         5.618INCREMENTAL CPU TIME, SECONDS        15.864TOTAL CPU TIME, SECONDS

CRUDE ITER NO.   5   DELTA PHI MAX =   5.109E-03   I =  37   J =   9   K =  16   NSP =   4
CIR =  0.         0.         0.         0.         -9.595E-01 -1.033E+00 -1.015E+00 -9.664E-01 -7.986E-01
CIR =  -5.311E-01
CIR =  0.         0.         0.         1.220E+00  8.361E-01  5.822E+00  5.522E+00  5.099E+00  4.637E+00  4.127E+00
CIR =  3.665E+00  3.364E+00  3.301E+00  3.184E+00  2.988E+00  2.695E+00  2.261E+00  1.678E+00

         1.406INCREMENTAL CPU TIME, SECONDS        17.270TOTAL CPU TIME, SECONDS
```

Figure 13. Flow solution convergence output

```
FINE ITER   6 DPM= 8.035E-03 I=  18 J=   9 K=   6 DPMAV= 1.898E-04 RSD= 2.265E+00 I= 518 J=  17 K=   6 RSDAV= 2.041E-02

CRUDE ITER NO.   6   DELTA PHI MAX =   4.216E-03   I =  36   J =   7   K =  20   NSP =   0
CIR =   0.          0.          0.          0.          0.         -8.883E-01  -9.882E-01  -9.832E-01  -9.333E-01  -8.224E-01
CIR =  -5.345E-01
CIR =   0.          0.          0.          1.220E+00   8.361E-01   5.456E+00   5.451E+00   4.901E+00   4.365E+00   4.069E+00
CIR =   3.554E+00   3.251E+00   3.167E+00   3.086E+00   2.950E+00   2.666E+00   2.180E+00   1.599E+00

FINE ITER   7 DPM= 4.155E-03 I=  17 J=   6 K=   6 DPMAV= 1.768E-04 RSD= 1.099E+00 I= 518 J=  17 K=   6 RSDAV= 2.501E-02

CRUDE ITER NO.   7   DELTA PHI MAX =   3.456E-03   I = 538   J =  10   K =  20   NSP =   0
CIR =   0.          0.          0.          0.          0.         -9.332E-01  -9.912E-01  -9.916E-01  -9.436E-01  -8.053E-01
CIR =  -5.461E-01
CIR =   0.          0.          0.          1.220E+00   8.361E-01   5.583E+00   5.438E+00   5.007E+00   4.522E+00   4.094E+00
CIR =   3.628E+00   3.326E+00   3.244E+00   3.144E+00   2.976E+00   2.687E+00   2.227E+00   1.659E+00

FINE ITER   8 DPM= 3.404E-03 I= 119 J=   6 K=  18 DPMAV= 1.690E-04 RSD= 9.144E-01 I=   3 J=  23 K=  18 RSDAV= 2.586E-02

CRUDE ITER NO.   8   DELTA PHI MAX =   2.790E-03   I =  38   J =   9   K =  20   NSP =   0
CIR =   0.          0.          0.          0.          0.         -8.964E-01  -9.647E-01  -9.759E-01  -9.303E-01  -8.099E-01
CIR =  -5.467E-01
CIR =   0.          0.          0.          1.220E+00   8.361E-01   5.417E+00   5.394E+00   4.956E+00   4.468E+00   4.082E+00
CIR =   3.611E+00   3.307E+00   3.219E+00   3.126E+00   2.970E+00   2.682E+00   2.215E+00   1.655E+00

FINE ITER   9 DPM= 3.563E-03 I= 118 J=   6 K=  17 DPMAV= 1.623E-04 RSD= 6.861E-01 I= 117 J=   9 K=  18 RSDAV= 2.420E-02

CRUDE ITER NO.   9   DELTA PHI MAX =   2.587E-03   I =  10   J =  13   K =  22   NSP =   0
CIR =   0.          0.          0.          0.          0.         -9.120E-01  -9.580E-01  -9.729E-01  -9.301E-01  -8.033E-01
CIR =  -5.492E-01
CIR =   0.          0.          0.          1.220E+00   8.361E-01   5.424E+00   5.366E+00   4.978E+00   4.509E+00   4.091E+00
CIR =   3.635E+00   3.332E+00   3.241E+00   3.141E+00   2.977E+00   2.688E+00   2.228E+00   1.669E+00

        13.929INCREMENTAL CPU TIME, SECONDS         31.199TOTAL CPU TIME, SECONDS


FINE ITER  10 DPM= 3.779E-03 I= 117 J=   6 K=  17 DPMAV= 1.508E-04 RSD= 6.418E-01 I= 117 J=   9 K=  18 RSDAV= 2.375E-02

CRUDE ITER NO.  10   DELTA PHI MAX =   2.478E-03   I =  10   J =  14   K =  22   NSP =   0
CIR =   0.          0.          0.          0.          0.         -8.894E-01  -9.390E-01  -9.607E-01  -9.222E-01  -8.024E-01
CIR =  -5.443E-01
CIR =   0.          0.          0.          1.220E+00   8.361E-01   5.325E+00   5.325E+00   4.960E+00   4.499E+00   4.091E+00
CIR =   3.639E+00   3.337E+00   3.239E+00   3.140E+00   2.977E+00   2.689E+00   2.229E+00   1.684E+00

         3.206INCREMENTAL CPU TIME, SECONDS         34.405TOTAL CPU TIME, SECONDS
```

Figure 13. (Concluded)

PRESSURE COEFFICIENTS (CP)

STATION 11 J= 11 2Y/B= .460 Y= 75.443 LOCAL CHORD= 84.648 CL (CIRCULATION) = .086

		UPPER SURFACE						LOWER SURFACE					
I	XWF	X/C	Y/C	CP	MACH	U	V	X/C	Y/C	CP	MACH	U	V
18	-3.315	.629	1.130	.396	.619	-.2114	-.1033	.629	-1.130	.900	.346	-.5402	-.2746
19	-2.250	1.887	1.918	-.123	.856	.0601	.0439	1.887	-1.918	.483	.577	-.2606	-.1171
20	-1.186	3.145	2.414	-.334	.952	.1578	.0953	3.145	-2.414	.269	.678	-.1393	-.0468
21	-.121	4.403	2.803	-.409	.988	.1917	.1119	4.403	-2.803	.154	.731	-.0779	-.0104
22	.944	5.660	3.122	-.455	1.010	.2120	.1209	5.660	-3.122	.040	.782	-.0205	.0230
23	2.009	6.918	3.390	-.471	1.018	.2196	.1215	6.918	-3.390	-.053	.824	.0251	.0477
24	3.073	8.176	3.619	-.471	1.018	.2201	.1172	8.176	-3.619	-.096	.843	.0460	.0561
25	4.138	9.434	3.820	-.484	1.024	.2260	.1155	9.434	-3.820	-.143	.865	.0684	.0651
26	5.203	10.692	3.996	-.492	1.027	.2301	.1121	10.692	-3.996	-.184	.883	.0878	.0723
27	6.268	11.950	4.152	-.498	1.031	.2334	.1080	11.950	-4.152	-.217	.898	.1034	.0773
28	7.332	13.208	4.288	-.504	1.034	.2367	.1038	13.208	-4.288	-.246	.912	.1172	.0814
29	8.397	14.465	4.408	-.509	1.036	.2393	.0997	14.465	-4.408	-.270	.923	.1284	.0845
30	9.462	15.723	4.514	-.506	1.034	.2384	.0940	15.723	-4.514	-.284	.929	.1352	.0851
31	10.527	16.981	4.607	-.506	1.034	.2388	.0907	16.981	-4.607	-.292	.933	.1390	.0844
32	11.592	18.239	4.687	-.506	1.034	.2387	.0898	18.239	-4.687	-.296	.935	.1411	.0833
33	12.656	19.497	4.757	-.479	1.021	.2267	.0818	19.497	-4.757	-.294	.934	.1402	.0807
34	13.721	20.755	4.817	-.441	1.003	.2102	.0712	20.755	-4.817	-.288	.931	.1377	.0775
35	14.786	22.013	4.867	-.417	.992	.1992	.0648	22.013	-4.867	-.284	.929	.1359	.0751
36	15.851	23.270	4.908	-.396	.982	.1896	.0592	23.270	-4.908	-.280	.927	.1339	.0729
37	16.915	24.528	4.942	-.376	.972	.1806	.0540	24.528	-4.942	-.274	.925	.1314	.0705
38	17.980	25.786	4.967	-.359	.964	.1731	.0496	25.786	-4.967	-.269	.922	.1291	.0681
39	19.045	27.044	4.985	-.347	.959	.1674	.0462	27.044	-4.985	-.265	.921	.1275	.0664
40	20.110	28.302	4.997	-.334	.953	.1615	.0427	28.302	-4.997	-.262	.919	.1261	.0649
41	21.174	29.560	5.002	-.319	.946	.1547	.0385	29.560	-5.002	-.257	.917	.1236	.0629
42	22.239	30.818	5.001	-.307	.940	.1488	.0351	30.818	-5.001	-.251	.914	.1208	.0608
43	23.304	32.075	4.994	-.297	.935	.1442	.0325	32.075	-4.994	-.245	.911	.1184	.0591
44	24.369	33.333	4.982	-.283	.929	.1377	.0286	33.333	-4.982	-.237	.908	.1145	.0564
45	25.433	34.591	4.965	-.264	.920	.1287	.0231	34.591	-4.965	-.225	.902	.1089	.0527
46	26.498	35.849	4.943	-.249	.913	.1215	.0192	35.849	-4.943	-.215	.898	.1043	.0500
47	27.563	37.107	4.916	-.237	.907	.1158	.0163	37.107	-4.916	-.208	.894	.1008	.0482
48	28.628	38.365	4.884	-.225	.902	.1100	.0130	38.365	-4.884	-.200	.891	.0972	.0460
49	29.692	39.623	4.849	-.216	.898	.1061	.0107	39.623	-4.849	-.195	.888	.0946	.0445
50	30.757	40.881	4.809	-.213	.897	.1045	.0103	40.881	-4.809	-.192	.887	.0936	.0443
51	31.822	42.138	4.765	-.207	.894	.1017	.0093	42.138	-4.765	-.188	.885	.0916	.0436
52	32.887	43.396	4.717	-.196	.889	.0962	.0059	43.396	-4.717	-.180	.881	.0876	.0412
53	33.951	44.654	4.666	-.186	.884	.0913	.0029	44.654	-4.666	-.172	.878	.0841	.0392
54	35.016	45.912	4.612	-.179	.881	.0882	.0013	45.912	-4.612	-.167	.876	.0818	.0380
55	36.081	47.170	4.554	-.170	.877	.0839	-.0013	47.170	-4.554	-.160	.873	.0785	.0361
56	37.146	48.428	4.493	-.157	.871	.0774	-.0056	48.428	-4.493	-.150	.868	.0733	.0328
57	38.210	49.686	4.428	-.146	.866	.0718	-.0091	49.686	-4.428	-.140	.864	.0689	.0302
58	39.275	50.943	4.362	-.138	.862	.0681	-.0113	50.943	-4.362	-.135	.861	.0661	.0286
59	40.340	52.201	4.292	-.132	.860	.0650	-.0132	52.201	-4.292	-.129	.859	.0636	.0274
60	41.405	53.459	4.219	-.127	.858	.0629	-.0143	53.459	-4.219	-.126	.857	.0620	.0267
61	42.469	54.717	4.145	-.125	.856	.0615	-.0146	54.717	-4.145	-.124	.856	.0609	.0266
62	43.534	55.975	4.067	-.120	.854	.0593	-.0157	55.975	-4.067	-.120	.854	.0589	.0258

Figure 14. Example of fine grid solution output

63	44.599	57.233	3.987	-.113	.851	.0557	-.0178	57.233	-3.987	-.113	.851	.0558	.0241
64	45.664	58.491	3.905	-.105	.847	.0518	-.0204	58.491	-3.905	-.107	.848	.0526	.0221
65	46.728	59.748	3.820	-.099	.845	.0490	-.0223	59.748	-3.820	-.102	.846	.0502	.0207
66	47.793	61.006	3.733	-.091	.841	.0450	-.0253	61.006	-3.733	-.095	.843	.0467	.0183
67	48.858	62.264	3.644	-.081	.837	.0397	-.0294	62.264	-3.644	-.085	.838	.0421	.0148
68	49.923	63.522	3.553	-.073	.833	.0359	-.0322	63.522	-3.553	-.078	.835	.0388	.0126
69	50.987	64.780	3.460	-.068	.831	.0333	-.0340	64.780	-3.460	-.074	.833	.0365	.0112
70	52.052	66.038	3.365	-.065	.829	.0315	-.0352	66.038	-3.365	-.070	.832	.0349	.0103
71	53.117	67.296	3.268	-.063	.828	.0305	-.0355	67.296	-3.268	-.068	.831	.0339	.0102
72	54.182	68.553	3.169	-.059	.827	.0287	-.0364	68.553	-3.169	-.065	.829	.0323	.0095
73	55.246	69.811	3.068	-.054	.825	.0264	-.0378	69.811	-3.068	-.061	.827	.0302	.0083
74	56.311	71.069	2.966	-.049	.822	.0235	-.0397	71.069	-2.966	-.055	.825	.0276	.0067
75	57.376	72.327	2.862	-.041	.819	.0197	-.0428	72.327	-2.862	-.049	.822	.0242	.0041
76	58.441	73.585	2.756	-.036	.816	.0171	-.0451	73.585	-2.756	-.044	.820	.0218	.0021
77	59.505	74.843	2.648	-.027	.812	.0125	-.0492	74.843	-2.648	-.036	.816	.0177	-.0014
78	60.570	76.101	2.538	-.017	.808	.0070	-.0540	76.101	-2.538	-.026	.812	.0128	-.0055
79	61.635	77.358	2.427	-.011	.805	.0040	-.0566	77.358	-2.427	-.020	.809	.0102	-.0076
80	62.700	78.616	2.314	-.007	.803	.0019	-.0583	78.616	-2.314	-.017	.808	.0083	-.0090
81	63.764	79.874	2.199	-.005	.802	.0007	-.0589	79.874	-2.199	-.015	.807	.0072	-.0095
82	64.829	81.132	2.082	-.002	.801	-.0006	-.0596	81.132	-2.082	-.012	.805	.0060	-.0101
83	65.894	82.390	1.963	.002	.799	-.0030	-.0611	82.390	-1.963	-.008	.804	.0038	-.0114
84	66.959	83.648	1.843	.007	.797	-.0057	-.0632	83.648	-1.843	-.003	.801	.0013	-.0132
85	68.023	84.906	1.720	.014	.794	-.0093	-.0663	84.906	-1.720	.004	.798	-.0019	-.0160
86	69.088	86.164	1.596	.022	.790	-.0135	-.0706	86.164	-1.596	.011	.795	-.0058	-.0198
87	70.153	87.421	1.471	.030	.787	-.0176	-.0751	87.421	-1.471	.018	.792	-.0094	-.0237
88	71.218	88.679	1.343	.045	.780	-.0263	-.0837	88.679	-1.343	.032	.785	-.0168	-.0309
89	72.283	89.937	1.214	.061	.772	-.0352	-.0923	89.937	-1.214	.047	.779	-.0242	-.0381
90	73.347	91.195	1.082	.069	.769	-.0396	-.0966	91.195	-1.082	.054	.776	-.0279	-.0416
91	74.412	92.453	.949	.074	.767	-.0422	-.0992	92.453	-.949	.058	.774	-.0302	-.0440
92	75.477	93.711	.814	.076	.766	-.0434	-.0999	93.711	-.814	.061	.773	-.0316	-.0450
93	76.542	94.969	.675	.087	.761	-.0496	-.1051	94.969	-.675	.075	.766	-.0389	-.0515
94	77.606	96.226	.535	.110	.750	-.0626	-.1167	96.226	-.535	.098	.756	-.0516	-.0632
95	78.671	97.484	.392	.137	.738	-.0779	-.1303	97.484	-.392	.126	.743	-.0666	-.0768
96	79.736	98.742	.248	.167	.725	-.0957	-.1462	98.742	-.248	.160	.728	-.0858	-.0944
97	80.801	100.000	.105	.204	.708	-.1183	-.1667	100.000	-.105	.203	.708	-.1109	-.1175

LOCAL SECTION CL (CP INTEG.) = .08340
LOCAL SECTION CM (CP INTEG.) = .01371
LOCAL SECTION CD (CP INTEG.) = .00476

Figure 14. (Concluded)

WING EXPOSED AREA = 20361.104

WING REFERENCE AREA = 30237.000

WING ASPECT RATIO = 3.56

WING TAPER RATIO = .33

WING MEAN AERODYNAMIC CHORD =101.375

WING AVERAGE CHORD = 92.186

X-POSITION FOR PITCHING MOMENT = .000

Z-POSITION FOR PITCHING MOMENT = 0.000

WING FORCE AND MOMENT COEFFICIENTS

WING CL	WING CM	WING CD
.04693	-.21111	.00061

WING PRESSURE DRAG = .00061

WING FRICTION DRAG = 0.00000

PITCHING MOMENT DUE TO DRAG = -.00018

CANARD FORCE AND MOMENT BASED ON SEXPC = 5431.315

CL	CM	CD(PRESSURE)	CD(FRICTION)
-.02768	.10143	-.00371	0.00000

CANARD FORCE AND MOMENT COEFFICIENTS

CANARD CL	CANARD CM	CANARD CD
-.00497	.01822	-.00067

Figure 15. Force and moment output

CANARD PRESSURE DRAG = -.00067

CANARD FRICTION DRAG = 0.00000

PITCHING MOMENT DUE TO DRAG = 0.00000

TOTAL FORCE AND MOMENT COEFFICIENTS

TOTAL CL	TOTAL CM	TOTAL CD
.04196	-.19290	-.00006

TOTAL PRESSURE DRAG = -.00006

TOTAL FRICTION DRAG = 0.00000

Figure 15. (Continued)

BODY LENGTH = 600.000

BODY WETTED AREA = 93000.300

BODY PROJECTED AREA = 29459.057

BODY MAX. CROSS-SECTIONAL AREA = 4300.840

X-POSITION ABOUT WHICH MOMENTS ARE COMPUTED = .000

BODY FORCE AND MOMENT COEFFICIENTS

BODY CL	BODY CM	BODY CD
.00819	-.03018	.00019

BODY PRESSURE DRAG = .00019

BODY FRICTION DRAG = 0.00000

Figure 15. (Concluded)

TOTAL CONFIGURATION COEFFICIENTS

CL	CM	CD
.05015	-.22307	.00013

SPANWISE DISTRIBUTIONS

WING STATIONS

	2Y/B	CLINT	CCL/CAV	CMLOC	CCM/CAV/MAC	CDINT	CCD/CAV	CFINT	CCF/CAV
1	0.00000	0.00000	.05470	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
2	.04193	0.00000	.06604	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
3	.08410	0.00000	.07745	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
4	.12673	0.00000	.08898	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
5	.17007	0.00000	.10071	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
6	.21436	.07394	.11269	.00583	-.52028	.01191	.01816	0.00000	0.00000
7	.25986	.07956	.11211	.00744	-.51773	.00396	.00558	0.00000	0.00000
8	.30686	.08579	.10733	.01035	-.49553	-.00518	-.00649	0.00000	0.00000
9	.35566	.08763	.09526	.01356	-.43993	-.00968	-.01052	0.00000	0.00000
10	.40659	.09023	.08649	.01455	-.40002	-.00122	-.00117	0.00000	0.00000
11	.46002	.08340	.07658	.01371	-.35014	.00476	.00437	0.00000	0.00000
12	.51638	.08022	.07025	.01243	-.31801	.00371	.00325	0.00000	0.00000
13	.57615	.08169	.06786	.00997	-.30499	.00890	.00739	0.00000	0.00000
14	.63991	.08532	.06677	.00826	-.29717	.00244	.00191	0.00000	0.00000
15	.70837	.08593	.06278	.00588	-.27637	.01061	.00775	0.00000	0.00000
16	.78239	.08523	.05738	.00553	-.24841	-.00106	-.00072	0.00000	0.00000
17	.86309	.08524	.05205	.00742	-.22046	-.02746	-.01677	0.00000	0.00000
18	.95193	.06329	.03428	-.00037	-.14374	.02506	.01358	0.00000	0.00000
19	1.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

CANARD STATIONS

	2Y/B	CLINT	CCL/CAV	CMLOC	CCM/CAV/MAC	CDINT	CCD/CAV	CFINT	CCF/CAV
1	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
2	.08597	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
3	.17241	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
4	.25980	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
5	.34864	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
6	.43943	-.02215	-.02139	.00498	.07938	.00469	.00453	0.00000	0.00000
7	.53271	-.02445	-.02156	.00325	.07873	-.00907	-.00800	0.00000	0.00000
8	.62907	-.02800	-.02148	.00187	.07807	-.00816	-.00626	0.00000	0.00000
9	.72910	-.03139	-.02034	.00075	.07419	-.00201	-.00130	0.00000	0.00000
10	.83351	-.03411	-.01786	-.00051	.06559	-.00075	-.00039	0.00000	0.00000
11	.94304	-.03216	-.01265	-.00102	.04700	-.00074	-.00029	0.00000	0.00000
12	1.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

Figure 16. Spanload output

*U.S.Government Printing Office: 1981 — 559-005/4026

DA
FILM